Universidade de Aveiro Departamento de Electrónica Telecomunicações e Informática
2009

Carlos Daniel Correia de Almeida

Integração de ferramentas de análise de dados laboratoriais

Universidade de Aveiro Departamento de Electrónica Telecomunicações e Informática
2009

Carlos Daniel Correia de Almeida

Integração de ferramentas de análise de dados laboratoriais

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Electrónica e de Telecomunicações, realizada sob a orientação científica do Doutor José Luís Guimarães Oliveira, Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Dedico este trabalho aos meus pais, aos meus irmãos e à minha namorada por todo o apoio e ajuda prestada durante a execução deste trabalho.

o júri

Presidente

Prof. Doutor Joaquim Arnaldo Carvalho Martins
Professor Catedrático da Universidade de Aveiro

Prof. Doutor Rui Pedro Sanches de Castro Lopes
Professor Coordenador da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Bragança

Prof. Doutor José Luís Guimarães Oliveira
Professor Associado da Universidade de Aveiro

agradecimentos

Agradeço a todos os colegas do grupo de bioinformática pela ajuda e companheirismo.
Agradeço aos meus irmãos, Pedro Almeida e António Almeida, e ao meu amigo Ricardo Pinto pelos debates, pela disponibilidade e pela ajuda prestadas ao longo da realização deste trabalho.
Agradeço às doutoras Maria José Correia e Marlene Barros, docentes na Universidade Católica Portuguesa Centro Regional das Beiras pelas preciosas explicações na área da biologia.
Para finalizar, um agradecimento muito especial ao meu orientador, Prof. Doutor José Luís Oliveira, e ao meu co-orientador Joel Arrais, pelo apoio, pela ajuda, pela paciência, pela tolerância e principalmente por acreditarem em mim.

palavras-chave WebServices, folhas de cálculo, Visual Studio Tools for Office, plataforma .NET

resumo

Existem muitas ferramentas que permitem consumir WebServices e analisar os dados por eles devolvidos. No entanto, nenhuma delas utiliza uma aplicação amplamente difundida na sociedade que esteja potencializada para tratar dados em estruturas tabulares.

As folhas de cálculo oferecem a possibilidade de analisar estas estruturas, criar gráficos ou realizar análise estatística de uma forma bastante simples. Por este motivo tornaram-se uma das ferramentas de trabalho mais populares na nossa sociedade.

Neste trabalho é apresentada uma solução que permite consumir WebServices e integrar os dados resultantes em folhas de cálculo. Assim, os investigadores poderão tirar partido de todas as funcionalidades das folhas de cálculo para realizarem o seu trabalho.

keywords WebServices, worksheets, Visual Studio Tools for Office,.NET framework

abstract WebServices is a common disseminated technology. Its consumption is typically performed by specially designed applications. However, these applications aren't prepared to deal with tabular structures.
Worksheets allow to analyze these structures and can easily create charts or make statistics analysis. For these reasons, researchers already use worksheets to help them to perform their tasks.
The main goal of this project is to provide new features of data integration to an existing tool that allows worksheets manipulation. These features will be developed bearing always in mind that the data integration must be easy to use.

Índice

Índice de figuras e tabelas

1 Introdução

1.1 Enquadramento

Nas últimas décadas a área da informática tem sofrido um grande desenvolvimento. Estão constantemente a surgir novas tecnologias que permitem aos programadores construir aplicações cada vez mais complexas e robustas. A maioria das aplicações desenvolvidas é normalmente utilizada para fins didácticos, educacionais ou como ferramentas de trabalho.

As ferramentas de trabalho são pensadas para ajudar o utilizador a realizar as tarefas rotineiras. Estas podem surgir para ajudar a resolver pequenos problemas ou para completar tarefas que dificilmente seriam realizadas por seres humanos.

Um dos objectivos em comum a estas aplicações é o de poupar tempo para que o trabalho que obrigatoriamente terá de ser realizado pelo ser humano tenha uma maior rentabilidade, quer a nível económico, quer a nível de pesquisa científica. Por outro lado, o ser humano tem tendência a cometer erros quando o tipo de trabalho em que está envolvido é demasiadamente repetitivo. A área da biologia reúne os critérios acima descritos e assim surge como uma das mais recentes áreas que necessita de ferramentas informáticas. A vontade do ser humano compreender a complexa organização dos seres vivos, descobrir novas patologias e a cura para as mesmas, veio promover o aparecimento destas novas ferramentas.

O ramo da biologia é muito complexo e é enorme a velocidade com que são geradas novas informações. Seria impossível ao ser humano tirar qualquer partido desta informação caso não existissem ferramentas informáticas ao seu dispor. Estas ferramentas podem ir desde o simples arquivo de informação em bases de dados (quer públicas, quer proprietárias) até ferramentas de extrema complexidade que fazem análise de dados experimentais para, por exemplo, detectar padrões de repetição em sequenciação de DNA.

A quantidade de dados que hoje em dia existe é surpreendente e existem diversas fontes de informação através das quais os investigadores podem realizar o seu trabalho. Acompanhando este enorme volume de dados, vêm por muitas vezes novas aplicações. No entanto, estas podem apresentar interfaces e estruturas que poderão ser radicalmente diferentes às anteriores. Este facto pode ser encarado pelos investigadores como um elemento desencorajador ao seu uso. Muitas vezes os investigadores olham para estas ferramentas como "mais uma aplicação" em que terão de despender parte do seu precioso tempo. Assim, surgiu a necessidade de criar uma nova ferramenta de trabalho que ao mesmo tempo fosse robusta e funcional. Por outro lado, a interface com o utilizador teria de ser simples de compreender e familiar aos utilizadores.

Normalmente, os dados que resultam da análise feita por investigadores podem ser facilmente apresentados em tabelas. Por vezes, ao analisar estes dados, surge a necessidade de realizar análise estatística e exportar esta mesma análise para novas tabelas ou gráficos. Poder-se-ia optar por construir uma nova aplicação, como uma nova interface. No entanto, iria surgir a necessidade de integrar ferramentas que permitissem efectuar cálculos, construir gráficos ou criar novas tabelas. Por estes motivos, surgiu a ideia de integrar os dados resultantes do consumo dos WebServices em folhas de cálculo, uma vez que esta ferramenta já tem todas estas funcionalidades. Não se conseguem construir aplicações com interfaces e capacidade de processamento tão desenvolvidas como é possível, por exemplo, no desenvolvimento de aplicações Windows. No entanto pode-se tirar partido de todas as funcionalidades de uma folha de cálculo.

1.2 Objectivos

Os dados que os investigadores utilizam para realizar as suas pesquisas estão muitas vezes organizados em estruturas tabulares. Desta forma, este trabalho tem como objectivo o desenvolvimento de ferramentas que permitam importar, de uma forma transparente para o utilizador, os dados resultantes do consumo de WebServices em folhas de cálculo. Assim, poder-se-ão utilizar todas as funcionalidades das folhas de cálculo para organizar e analisar dados biológicos.

Será feita uma análise dos softwares existentes que permitem manipular folhas de cálculo e das tecnologias que permitem adaptar estes softwares às necessidades dos utilizadores. Posteriormente, optar-se-á pelo software que tenha possibilidade de ser alterado e que se encontre mais difundido na sociedade.

Como foi referido anteriormente, o acesso a dados será feito a partir do consumo de WebServices. Os métodos disponibilizados por estes serviços Web acedem à base de dados do GeNS ("Genomic Name Server") e retornam vários tipos de dados que serão integrados em folhas de cálculo para posterior análise. A ferramenta desenvolvida deverá ter duas interfaces. Uma gráfica em que a partir do clique em botões os dados sejam introduzidos em folhas de cálculo. A outra interface deverá permitir que os utilizadores usem a ferramenta tal como as funções já existentes em folhas de cálculo, como por exemplo, a função "=SOMA (A1,B1) ". Sempre que existirem desafios na integração de dados, tentar-se-á sempre aproximar a interface ao modo habitual de utilização das folhas de cálculo.

Espera-se que esta ferramenta venha ajudar os investigadores no seu trabalho rotineiro e que as ferramentas desenvolvidas permitam extrair informação sobre dados existentes em bases de dados. É esperado que se consigam relacionar vários tipos de dados relativos a organismos, genes, proteínas e entidades biológicas e apresentá-los em folhas de cálculo.

1.3 Estrutura da dissertação

Para além deste e dos capítulos de bibliografia e anexos, este documento divide-se em mais três capítulos.

No capítulo 2 foi analisado qual o melhor software para desenvolver a ferramenta de integração de dados. Posteriormente foi feita uma análise da tecnologia que foi usada para manipular folhas de cálculo. Uma vez que os WebServices são também um ponto fundamental deste trabalho, será também feita uma revisão sobre esta tecnologia.

No capítulo 3 será apresentado o enquadramento deste trabalho e as razões que motivaram à sua realização. Para além disso, será feita uma descrição e análise das funcionalidades implementadas e dos desafios encontrados ao longo do desenvolvimento da aplicação.

No capítulo 4 é feita uma análise entre os objectivos que foram propostos e os atingidos. Para além disso será feita uma crítica global ao trabalho.

2 Tecnologias de suporte

Na área das ciências biológicas existem muitas instituições que armazenam os dados das suas pesquisas em bases de dados. Algumas destas instituições partilham os resultados das suas pesquisas ao resto da comunidade científica. Por muitas vezes, o acesso a estes dados é feito a partir do consumo de WebServices.

Os WebServices permitem disponibilizar um conjunto de funções que podem ser consumidas quer a partir de um navegador Web, quer por via programática. A possibilidade de serem facilmente acedidos por via programática é um dos elementos que melhor diferencia WebServices de WebSites.

As ferramentas informáticas existentes produzem grandes volumes de dados que terão de ser analisados por outras ferramentas informáticas ou por investigadores. Os investigadores serão muitas vezes responsáveis por validar os dados produzidos pelas ferramentas informáticas. Esta razão promove a partilha da informação que se produz e os WebServices facilitam a troca de informação entre diferentes instituições. Com esta partilha de informação, é possível que pequenos grupos de investigadores possam realizar investigação na área da biologia sem que para tal necessitem de investir em equipamentos extremamente dispendiosos.

No seu dia-a-dia, os investigadores utilizam ferramentas que lhes disponibilizam o acesso a dados gerados por outras instituições. Estes dados tanto podem ser obtidos por ferramentas informáticas como através de um navegador Web. Muitas vezes estes dados apresentam uma característica em comum que é a possibilidade de serem apresentados em tabelas. Por vezes, os investigadores procuram utilizar ferramentas às quais já estejam familiarizados para procederem à análise destes dados. Assim, é normal que as folhas de cálculo já sejam uma das ferramentas de trabalho utilizadas pelos investigadores.

Segundo um estudo realizado pela empresa *"ClickStream Technologies"* entre Maio e Novembro de 2008 [1], a Microsoft apresenta um claro domínio de mercado no que diz respeito à utilização de programas que permitem manipular documentos de texto ou folhas de cálculo. A agência de notícias CNET publicou em 13 de Março de 2009 uma pequena revisão deste estudo[1]. Neste, concluiu-se que 51% dos utilizadores de internet residentes nos Estados Unidos da América utilizam o Microsoft Office.

Obtiveram-se também termos de comparação entre OpenOffice, Google Docs e Google Spreadsheets e Microsoft Word e Excel. Em 68% dos casos, os utilizadores das ferramentas Google utilizaram pelo menos uma vez o Microsoft Word. Isto revela que apesar de usarem esta ferramenta, os utilizadores têm a necessidade de usar uma ferramenta disponibilizado pela Microsoft para completarem o seu trabalho. Isto significa que a maioria dos utilizadores não consegue encarar as ferramentas Google como uma aplicação que poderia existir sozinha no mercado. Para o caso do OpenOffice os dados não são tão preocupantes. Apenas 26% dos utilizadores de OpenOffice necessitaram de utilizar ferramentas Microsoft para completarem o seu trabalho. Assim, o OpenOffice apresenta-se como o principal concorrente da Microsoft no mercado de aplicações catalogadas como ferramentas de trabalho.

O estudo anterior comprova que o Microsoft Office é sem dúvida líder no mercado de ferramentas de trabalho. Assim, o Microsoft Excel apresenta-se como um dos produtos informáticos mais divulgados na nossa sociedade e talvez seja esta a principal razão pela qual os investigadores já o utilizam para organizar e manipular alguns dados resultantes das suas pesquisas. A própria forma como muitas vezes os dados estão organizados convidam o uso de uma ferramenta que permita, de uma forma simples, apresentar e manipular estruturas tabulares. No entanto, os investigadores necessitavam sempre de utilizar outras ferramentas ou aceder a páginas Web para obter os dados que desejavam integrar no Excel. Desta forma, neste trabalho irá ser desenvolvida uma ferramenta que permitirá integrar directamente em folhas de cálculo dados que sejam provenientes de WebServices. Optou-se por utilizar o Excel para integrar os dados provenientes de WebServices.

[1] http://news.cnet.com/8301-13505_3-10195845-16.html

Uma vez que se optou por utilizar o Microsoft Excel, a ferramenta e suas interfaces terão que ser desenvolvidas utilizando também uma tecnologia da Microsoft. Desta forma decidiu-se utilizar uma tecnologia pertencente à plataforma dotNET[2] uma vez que esta é uma das mais recentes e poderosas tecnologias desenvolvidas pela Microsoft. Para além disso, dotNET é uma tecnologia que se encontra em amplo desenvolvimento.

Neste capítulo irão ser abordadas as principais tecnologias que permitem o desenvolvimento da ferramenta para o Excel e alguns aspectos essenciais à compreensão, utilização e implementação de WebServices. Também serão apresentadas algumas das principais características da plataforma dotNET uma vez que esta plataforma é a base da tecnologia que foi seleccionada para desenvolver este trabalho.

2.1 WebServices

O desejo de poder executar aplicações remotamente surgiu desde os primeiros dias da criação da Web. Os WebServices podem ser considerados como uma das respostas a este desejo. Estes podem ser encarados como aplicações que irão ser consumidas quer por outras aplicações Web, quer por aplicações Windows. Os WebServices podem fornecer *meta* dados que devem descrever as mensagens que eles consomem bem como o serviço que disponibilizam (WSDL – *"Web Service Description Language"*). Em dotNET, um WebService poderá ser visto como uma página ASP.NET que retorna aos seus clientes XML em vez de HTML.

Como o próprio nome indica, os WebServices são desenhados para oferecer serviços através da Web. Os WebServices não são uma tecnologia exclusiva da plataforma dotNET. Deste modo, existe, por exemplo, a possibilidade de através de WebServices disponibilizar o acesso a uma base de dados que poderá ser utilizado por várias plataformas diferentes. Assim, garantir-se-á que diferentes aplicações desenvolvidas em diferentes plataformas poderão partilhar o mesmo acesso a uma base de dados. Na verdade, o recorrente uso de

[2] Neste documento, pelas dificuldades de representação e leitura do nome dado pela Microsoft à plataforma .NET, optou-se por utilizar o nome dotNET em vez de .NET. Desta forma, garantir-se-á que o nome da plataforma irá ser lido de igual forma pelos leitores deste documento.

WebServices apenas se tornou possível porque diferentes empresas chegaram a um acordo da forma de invocar estes serviços.

Os métodos fornecidos pelos WebServices são invocados utilizando XML codificado. Os consumidores comunicam com os serviços através de um protocolo chamado SOAP (*"Simple Object Access Protocol"*). O protocolo SOAP é uma formalização XML para comunicação que tem por base a troca de mensagens. Este define a formatação das mensagens, a forma como as mensagens devem ser enviadas via HTTP e ainda um padrão de representação de erros [2]. Este protocolo permite que sejam enviados pela rede Web complexas estruturas de dados. Para tal, será realizada uma serialização dos dados para permitir que estes sejam transportados pela rede Web. Na aplicação receptora dos dados, será realizada uma desserialização do formato SOAP XML para o formato original. O diagrama da Figura 2.1 ajuda a compreender o processo de serialização e desserialização. É importante referir que apenas são serializados os campos e propriedades que estão definidos como sendo públicos.

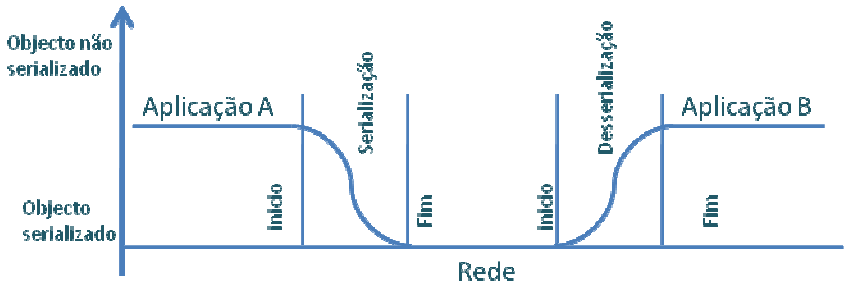


Figura 2.1 - Serialização de objectos

2.2 Modelo de objectos do Excel

A Figura 2.2 apresenta um modelo simplificado da hierarquia de objectos do Excel. A compreensão desta hierarquia é essencial para que se conseguiam manipular as folhas de cálculo por via programática ou mesmo para utilizar a interface do Excel. Trata-se de uma estrutura simples e intuitiva. Qualquer utilizador que esteja familiarizado com o Excel rapidamente se apercebe que dentro da aplicação do Excel existem folhas de cálculo e que

nestas folhas de cálculo existem células. As células são o elemento atómico das folhas de cálculo e é nelas que os utilizadores inserem valores ou fórmulas. As funções utilizam os valores existentes nestas células para realizar cálculos ou construir gráficos.

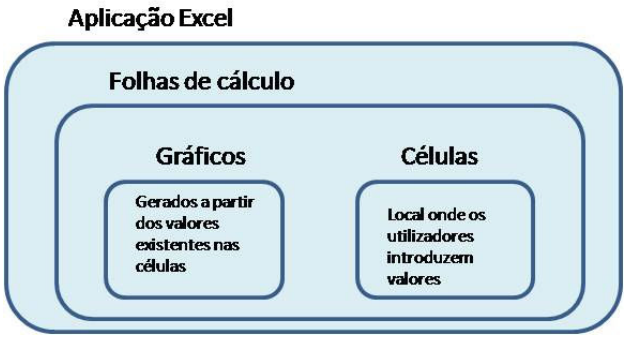


Figura 2.2 - Representação de uma simplificação do modelo de objectos do Excel [3]

2.3 Visual Studio Tools for Office

Visual Studio Tools for Office (VSTO) surgiu para satisfazer necessidades empresariais na construção de software baseado na tecnologia do Office. VSTO é uma tecnologia que permite aos documentos do Microsoft Office executar código contido num *assembly* dotNET. Esta não é a única tecnologia que permite adicionar novas funcionalidades ao Office. Previamente, os desenvolvedores de páginas Web usavam ASP clássico, *COM Interops* e *Visual Basic for Applications* (VBA) para desenvolver este tipo de aplicações. VSTO apenas veio facilitar o desenvolvimento de aplicações para Office [3] e tornar estas aplicações mais fiáveis.

VSTO foi reformulado para permitir a sua integração na plataforma dotNET (Figura 2.3). O redesenho da arquitectura veio promover a integração de objectos dotNET no Excel, no Word ou qualquer aplicação Office. A integração destes novos objectos veio oferecer a possibilidade de desenvolver novas aplicações que permitam adicionar novas funcionalidades às já existentes no Microsoft Office. Deste modo, poder-se-á adaptar cada uma destas aplicações às necessidades de cada utilizador.

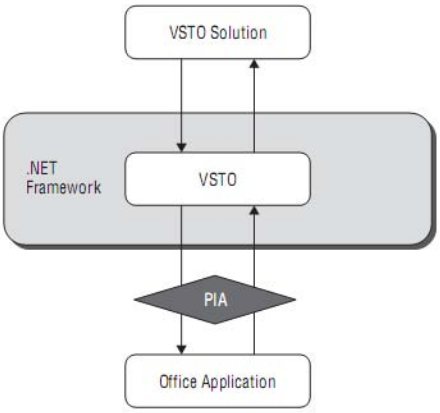


Figura 2.3 - Ilustração da comunicação entre VSTO e o Office [4]

Estes novos objectos dotNET vieram resolver muitos problemas de manuseamento de memória existentes no desenvolvimento de componentes *Microsoft Office Component Object Modeler* (COM).

A primeira versão de VSTO surgiu com o Visual Studio 2003. Desenvolver aplicações nas anteriores tecnologias (COM e VBA) requeria muito esforço por parte dos desenvolvedores de aplicações para Office e eram utilizados todos meios (macros, *hacks e Interop*) para garantir a estabilidade e o desempenho das aplicações construídas [3]. No entanto, existe muito software malicioso que está associado a macros e *hacks* o que muitas vezes pode levar a problemas de segurança do sistema informático. A Microsoft espera resolver alguns destes problemas de segurança através de VSTO uma vez que é possível, através do uso desta tecnologia, construir novas aplicações para o Office sem recorrer a macros. O código que é executado atrás do documento Office pode ter restrições de segurança aplicadas e deste modo poder-se-á sempre verificar a autenticidade do programa que está a ser executado, garantindo assim a segurança da aplicação.

Através do uso de VSTO podem ser construídos vários tipos de aplicações com formulários já definidos [5]. Entre vários destacam-se dois:

- Personalizações a nível do documento
- Novas funcionalidades para a aplicação (*add-ins).*

Ao utilizar os modelos disponibilizados pelo Visual Studio 2008 para construir personalizações a nível do documento, será criado um *assembly* que irá estar ligado a um documento individual. O documento poderá sofrer inúmeras alterações e podem-se integrar inúmeros objectos dotNET a estes documentos que vão desde botões e caixas de texto a objectos de acesso a bases de dados. No entanto estas alterações estarão sempre associadas apenas a um documento.

Os *add-ins* são novas funcionalidades que irão estar disponíveis a nível da aplicação. Estas funcionalidades poderão estar definidas para alterar o documento que está aberto na aplicação. Isto significa que poderá ser construído um vasto conjunto de métodos que podem proceder à personalização de vários documentos.

A Figura 2.4 apresenta um conjunto de botões que podem ser utilizados no desenvolvimento de *add-ins*. Cada um destes controlos poderá então ser programado em dotNET para realizar a função desejada.

Figura 2.4 - Exemplo de um add-in para o Excel

Na Figura 2.3 aparece descrito um módulo que ainda não foi referido neste documento. PIA é um acrónimo de "*Primary Interop Assemblies*". É através deste módulo que se torna possível utilizar as funcionalidades do Office através da tecnologia VSTO [6]. O Visual Studio adiciona automaticamente as referências necessárias para que se possa completar o projecto. Só existirá necessidade de integrar novas referências caso se desejem utilizar funcionalidades de outra aplicação Office que não aquela para a qual o projecto está a ser desenvolvido. Por exemplo, poderá haver a necessidade de utilizar ferramentas do Excel num projecto que foi iniciado como um projecto Word.

2.3.1 Desvantagens de VSTO

Obrigatoriedade da instalação da plataforma dotNET

Um dos principais problemas de aplicações construídas na tecnologia VSTO prende-se ao facto desta necessitar da instalação da plataforma dotNET. Em muitas empresas, os utilizadores não possuem permissões para instalar aplicações de software nem plataformas nos computadores. Por este motivo, os programadores poderão construir aplicações noutras tecnologias para superar esta limitação.

Segurança

Para se implementar segurança em aplicações desenvolvidas com a tecnologia VSTO, ter-se-ão de usar os mecanismos disponibilizados pela plataforma dotNET. Deste modo, o programador terá de conhecer detalhadamente alguns aspectos de segurança da plataforma dotNET para aplicar segurança à aplicação desenvolvida.

Desempenho

A instanciação da plataforma dotNET provoca atraso no inicio da aplicação uma vez que o código é compilado apenas no momento da instância ("Just In Time" - JIT) o que inevitavelmente irá prejudicar a desempenho. Outro problema que influencia o desempenho é o facto das aplicações construídas com base em VSTO terem de atravessar pequenas camadas de automação que envolvem acesso a objectos Microsoft Office COM. A construção optimizada destes objectos com VBA permite aumentar o desempenho uma vez existe uma maior aproximação de VBA à programação de objectos Office [3].

Uma vez que VSTO é uma tecnologia que pertence à plataforma dotNET achou-se relevante apresentar neste documento uma pequena revisão das principais características da plataforma dotNET.

2.4 Plataforma dotNET

A plataforma dotNET é um produto da Microsoft que foi lançado pela primeira vez no início de 2002. Actualmente encontra-se na versão 3.5. Esta plataforma tem vários objectivos em que alguns deles serão destacados neste documento.

Após a sua primeira versão, dotNET sofreu várias alterações e a cada nova versão, esta rica plataforma, apresenta novas funcionalidades em que um dos principais objectivos é simplificar a programação penalizando o mínimo possível o desempenho das aplicações desenvolvidas. Deste modo a Microsoft tenta cada vez mais permitir que a programação se aproxime da linguagem humana.

Na plataforma dotNET todos os objectos construídos e suportados pela mesma são derivados da classe *"System.Object"*. Desta forma é possível desenvolver a plataforma sem alterar radicalmente o funcionamento da mesma. As novas funcionalidades não irão alterar o modo de funcionamento das antigas o que garante que as aplicações construídas não têm que ser adaptadas caso seja lançada uma nova versão da plataforma.

Com esta nova plataforma, a Microsoft conseguiu resolver alguns problemas antigos de compatibilidade entre as aplicações e o sistema operativo. Se a plataforma dotNET for usada para construir aplicações, é possível garantir que as aplicações desenvolvidas poderão ser utilizadas em qualquer sistema operativo que tenha a plataforma instalada. Assim, as aplicações serão quase que independentes do sistema operativo.

A Figura 2.5[3] representa de uma forma esquemática parte da estrutura da plataforma dotNET. Como se pode perceber, primeiro ter-se-á de escolher uma linguagem dotNET para construir a aplicação. Através desta linguagem poder-se-ão então construir aplicações Windows ou Web. As aplicações desenvolvidas poderão possuir acessos a bases de dados que são conseguidos através da camada ADO.NET. Todo o código construído será, depois de compilado, interpretado pelo "*Common Language Runtime*" que será responsável em comunicar com o sistema operativo.

[3] http://www.devtopics.com/what-is-net/

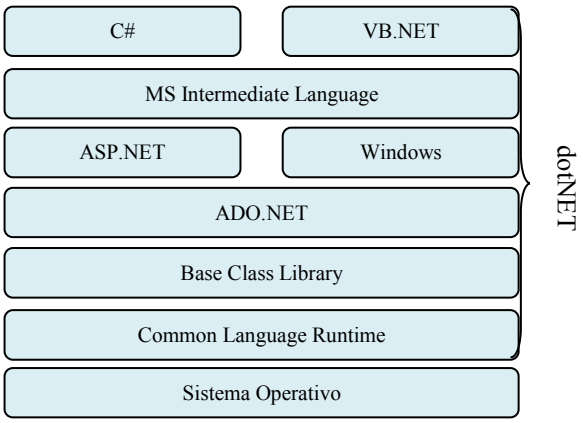


Figura 2.5 - Representação da plataforma dotNET

O núcleo da plataforma dotNET é o seu ambiente de execução, também conhecido como *"Common Language Runtime"* (CLR) [4]. O código de um programa que esteja a ser executado e controlado pelo CLR é também chamado de "*managed code"* (código manejável).

Qualquer programa que seja desenvolvido em C#, ou outra linguagem dotNET, antes de ser executado pelo CLR terá de ser compilado. São necessários dois passos para realizar a compilação:

- Compilação do código fonte para "*Microsoft Intermediate Language"* (IL)
- Compilação do IL para código nativo para ser interpretado pelo CLR.

Estes dois passos de compilação são cruciais uma vez que a existência de IL ("*managed code"*) é uma das principais chaves para o sucesso da plataforma dotNET. Na Figura 2.6 é apresentado um esquema da compilação de código fonte em código nativo. Embora apenas esteja representado o código C#, o processo é idêntico para qualquer das linguagens suportados pela plataforma dotNET.

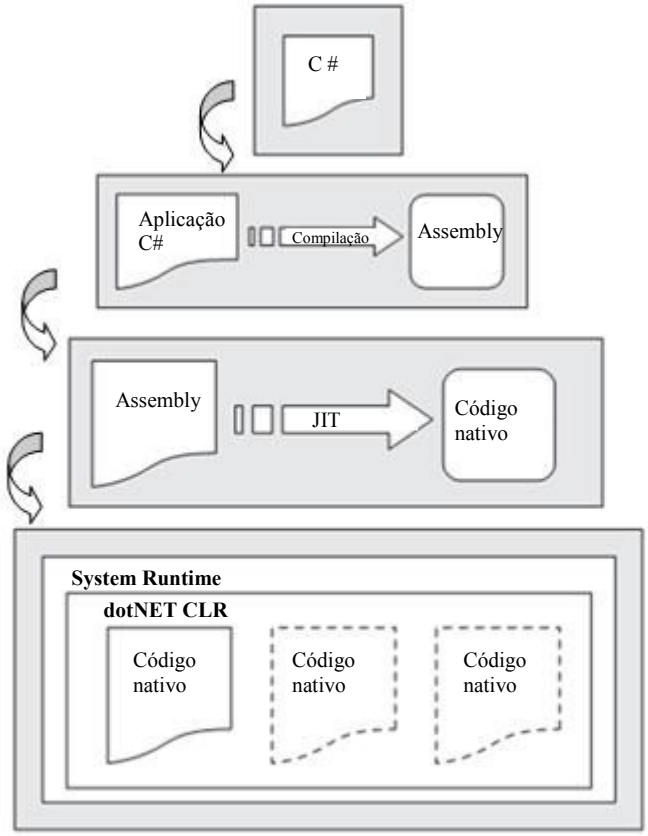


Figura 2.6 – Compilação de código na plataforma dotNET [7]

2.4.1 Independência de plataforma

Independência de plataforma significa que o mesmo ficheiro com o mesmo tipo de instruções possa ser executado em diferentes sistemas operativos. Em execução, o passo final da compilação deverá ser facilmente obtido para que o programa desenvolvido possa ser executado no ambiente em que se encontra presente. Apesar da Microsoft ter desenvolvido a plataforma dotNET apenas para o sistema operativo Windows é possível encontrar outras versões gratuitas desta plataforma, como por exemplo a Mono, que permitem utilizar a plataforma dotNET em Linux ou Mac OS. No entanto, não é possível garantir que as funcionalidades existentes em Mono correspondam totalmente às funcionalidades existentes na plataforma dotNET desenvolvida pela Microsoft.

2.4.2 Interoperabilidade de linguagem

O uso de "*managed code"* para além de estar associado à independência de plataformas também permite a interoperabilidade de linguagem. Isto significa que o código IL compilado por uma linguagem dotNET é compatível com o código IL compilado por outra linguagem de programação diferente [8]. Isto permite, por exemplo, a construção de um Web Site em que páginas diferentes possam ser construídas em linguagens dotNET diferentes [9].

Existem softwares, como por exemplo o "*Reflector*"[4], que permitem obter parte do código desenvolvido mesmo depois do projecto estar compilado num ficheiro executável ou num ficheiro ".dll". Estes softwares tentam realizar o processo inverso da compilação que a plataforma dotNET realiza. A partir do IL, tentam obter o código fonte desenvolvido. Uma vez que o código IL é comum às várias linguagens dotNET, é possível transformar o código nativo em VisualBasic.NET mesmo que o programa inicial tenha sido desenvolvido em C#.

Desta forma, é importante referir que devem ser tomadas medidas de protecção do código desenvolvido caso a aplicação a ser desenvolvida seja proprietária e cuja tecnologia envolvida seja de elevada confidencialidade.

2.4.3 Assemblies

Após o primeiro passo da compilação de uma aplicação, o código IL gerado é guardado num *assembly*. Os ficheiros resultantes deste processo tanto podem ser ficheiros e aplicações Windows (qualquer ficheiro com extensão ".exe") ou bibliotecas (ficheiros ".dll") que podem ser executados a partir de outra aplicação. Para além de código IL, os *assemblies* também contêm informação "*meta*", isto é, informação sobre a própria informação que o *assembly* possui, ou ainda recursos utilizados pelo IL, como por exemplo ficheiros áudio ou imagens. Esta informação ***"****meta"* torna possível que o *assembly* seja

[4] Http://www.red-gate.com/products/reflector/

completamente auto-descritivo. Isto permite que, por vezes, a instalação aplicações seja tão simples como copiar ficheiros para um computador remoto, com o único pressuposto que esse computador tem previamente instalada a plataforma dotNET [4].

2.4.4 Garbage Collector

O *Garbage Collector* é responsável por libertar memória de uma determinada aplicação assim que esta já não está a ser usada [10]. Anteriormente era da responsabilidade do programador fazer esta gestão de memória. Por vezes, pequenos erros de código resultavam em alocação de memória desnecessária e numa zona de memória errada [4] . Desta forma a aplicação poderia ficar progressivamente mais lenta e consequentemente poderia levar a erros no sistema.

Esta ferramenta inspecciona a memória do computador e remove tudo o que a aplicação já não está a necessitar [11]. Não é possível prever o intervalo de tempo entre duas execuções desta ferramenta. Tanto poderá ser executada milhares de vezes num segundo como uma única vez ao longo de vários segundos. Quanto maior for a necessidade de libertar memória, menor será o intervalo de tempo em duas execuções seguidas desta ferramenta.

2.4.5 Desempenho

A *Microsoft Intermediate Language (IL)* é sempre compilada em tempo real ("*Just in Time compiled"*, JIT). Em vez que compilar toda a aplicação de uma vez só, o que poderia levar a um significativo atraso no inicio da aplicação, o compilador JIT compila cada pedaço de código à medida que este vai sendo chamado. Depois de ser compilado uma primeira vez, o código nativo resultante é guardado até que se feche a aplicação evitando assim a necessidade de voltar a compilar este pedaço de código. A Microsoft defende que este processo é mais eficiente do que compilar toda a aplicação uma vez que é muito frequente que grandes blocos de código de qualquer aplicação não sejam executados e desta forma esse código nunca será compilado [4].

Esta é a razão pela qual se pode esperar que a execução de código manejável IL seja quase tão rápida como a execução de código máquina. No entanto não explica por que razão a Microsoft espera melhorias no desempenho. A explicação para este ponto prende-se ao facto do passo final da compilação ocorrer em tempo de execução e, neste momento, o compilador JIT saberá exactamente o tipo de processador em que o programa será executado e os recursos disponíveis para a execução da aplicação. Deste modo o código executável poderá ser optimizado para tirar partido de qualquer propriedade em específico que o sistema operativo possa fornecer [4]. Os compiladores tradicionais optimizam código, mas não o podem fazer para um processador em particular. Isto verifica-se uma vez que os compiladores tradicionais compilam o código antes de este ser enviado para o utilizador final. Por exemplo, não seria possível a estes compiladores prever se o código seria executado em processadores compatíveis com X86 ou processadores Alpha.

A Figura 2.7 representa um diagrama com a estrutura da plataforma dotNET. Através desta figura é mais uma vez possível observar a clara separação dos dois momentos de compilação do código desenvolvido:

- Compilação do código fonte para "*Microsoft Intermediate Language"* (IL)
- Compilação do IL para código nativo para ser interpretado pelo CLR.

2.4.6 Isolated Storage

A plataforma dotNET possui várias formas de aceder e monitorizar dados existentes no disco rígido.

Isolated Storage é uma tecnologia que poderá ser utilizada quando é necessário guardar ficheiros sem que o programador se tenha de preocupar se o utilizador final tem ou não permissões para manipular ficheiros no sistema [12]. É habitual que os utilizadores não tenham muitos privilégios uma vez que isto só tem vantagens a nível administrativo. Quanto menores forem os privilégios, maiores são as probabilidades de evitar que vírus ou spywares se infiltrem no sistema informático [13]. Existem vários cenários em que se pode utilizar esta funcionalidade [14]. Os dados são tipicamente guardados nas definições locais de cada utilizador. No entanto o endereço físico depende do sistema operativo utilizado.

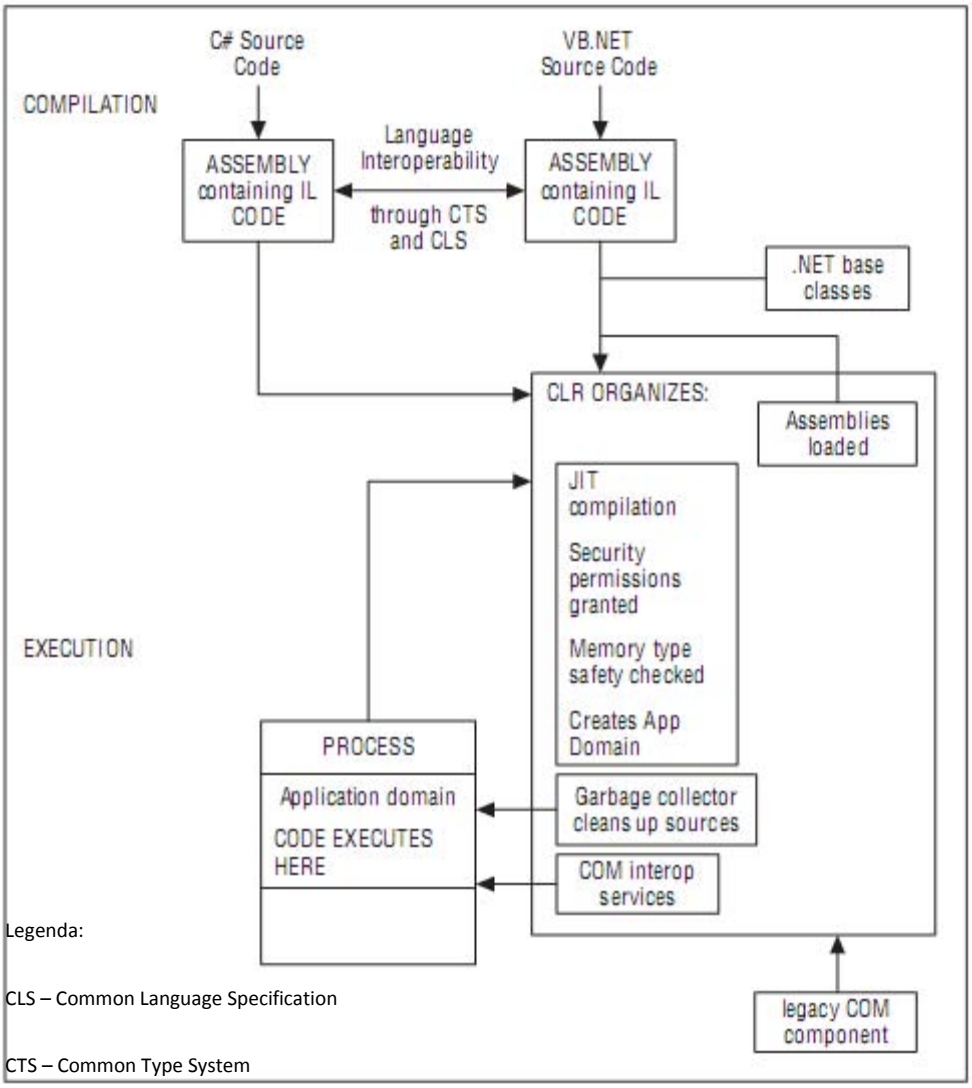


Figura 2.7 - Diagrama com as principais características da plataforma dotNET [4]

2.5 Sumário

Neste capítulo foram apresentados alguns conceitos sobre WebServices e as razões que promovem o uso destes mesmos serviços. Além dos WebServices foi feita uma apresentação de Visual Studio Tools for Office (VSTO). VSTO é a mais recente tecnologia da Microsoft que permite adaptar o Office às necessidades de cada utilizador. VSTO é uma tecnologia que pertence à família dotNET. Foram ainda apresentadas algumas das principais funcionalidades e características da plataforma dotNET uma vez que se verificou que não fazia sentido utilizar uma tecnologia sem que se conhecessem as suas bases.

Conceitos a reter sobre WebServices:

- São versáteis uma vez que permitem aceder a dados a partir da Web ou por via programática
- Podem ser utilizados a partir de várias plataformas
- Utilizam o protocolo HTTP para transmitir os dados
- WebServices são uma boa solução para aceder a bases de dados.

Conceitos a reter sobre VSTO:

- Permite criar adaptações a nível do documento e a nível da aplicação
- Permite integrar ferramentas desenvolvidas em dotNET nas aplicações do Office
- Permite manipular por via programática as aplicações do Office.

Conceitos a reter sobre a plataforma dotNET:

- É um produto da Microsoft que permite construir aplicações Windows ou Web.
- É uma plataforma em amplo desenvolvimento. Futuros desenvolvimentos não vão alterar o comportamento actual da plataforma.
- Independência do sistema operativo utilizado, embora a Microsoft apenas tenha desenvolvido a plataforma dotNET para sistemas operativos Windows.
- As aplicações construídas na plataforma dotNET adaptam-se ao hardware existente na altura da execução.

3 Integração de WebServices em Excel

Os investigadores têm ao seu dispor um vasto conjunto de bancos de dados que usam diariamente como fonte de informação para as suas pesquisas. Existe um grande esforço de várias instituições em estudar e desenvolver sistemas que permitam juntar os dados existentes nas diversas bases de dados num único sistema. Para além da centralização da informação, estes sistemas procuram também eliminar os dados que se encontrem repetidos.

O GeNS[5] (*"Genomic Name Server"*) trata-se de um sistema, desenvolvido pelo grupo de bioinformática da Universidade de Aveiro, que integra dados existentes em diferentes bases de dados num único sistema. O GeNS integra informação proveniente de WebServices, WebCrawlers, bases de dados e ficheiros tabulares. A Figura 3.1 apresenta um diagrama da estrutura do GeNS.

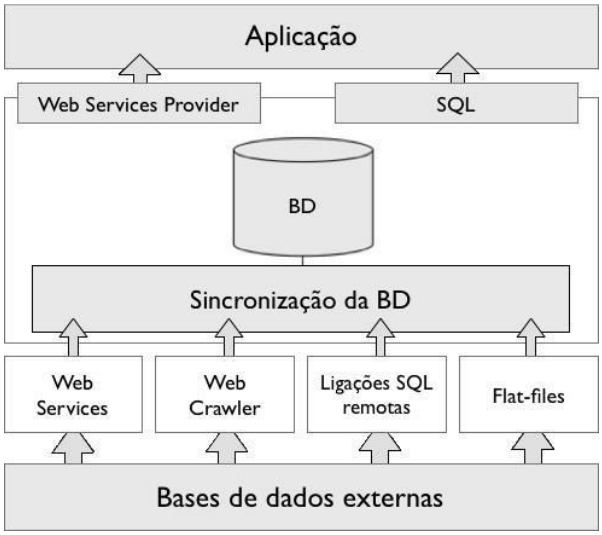


Figura 3.1- Diagrama representativo do sistema GeNS [15]

[5] http://bioinformatics.ua.pt/applications/gens

A camada de sincronização dos dados é responsável por fazer a triagem dos dados que serão introduzidos na nova base de dados. Assim, torna-se possível eliminar dados repetidos e integrar os dados únicos das diversas fontes de informação na nova base de dados.

O GeNS foi construído tendo sempre em consideração que a informação nele contida poderia ser consumida e analisada por outras ferramentas de trabalho. Desta forma o sistema permite que utilizadores acedam aos dados através de SQL ou de WebServices. Para usar SQL, os utilizadores terão de instalar uma cópia da base de dados na sua máquina local. Para evitar este ponto foram construídos WebServices que permitem o acesso à grande maioria dos dados presentes na base de dados.

O trabalho apresentado neste documento consiste na implementação de uma ferramenta de trabalho que irá preferencialmente utilizar o GeNS para integrar dados laboratoriais provenientes de WebServices em folhas de cálculo. As novas funcionalidades construídas permitirão que o Excel seja utilizado como uma aplicação pertencente à camada "*Application Layer*". Os métodos existentes no WebService disponibilizado pelo GeNS não eram os suficientes e desta forma foram também desenvolvidos novos métodos para permitirem o acesso a uma maior quantidade de dados.

Apesar do GeNS ser um sistema que possui imensa informação, a aplicação construída permitirá consumir WebServices que não pertencem ao GeNS. Este consumo não será o principal objectivo deste trabalho. Desta forma apenas será construído um método que através do URL completo dos métodos dos WebServices permita importar o XML que resulta da execução destes métodos. Assim, caso os investigadores o desejem, poderão adicionar dados de outras fontes de dados que não o GeNS para o Excel.

Uma vez que já existem vários WebServices (públicos e privados) e alguns sistemas de integração de dados (GeNS ou Taverna[6]) optou-se por desenvolver uma aplicação que utilizasse como base estes recursos para integrar dados em folhas de cálculo. Assim, as potencialidades das folhas de cálculo aliadas ao sistema de integração de dados tornarão esta aplicação única e funcional.

[6] http://taverna.sourceforge.net/

O software mais utilizado que permite manipular folhas de cálculo é o Microsoft Excel. É até comum que os utilizadores conheçam o Excel como sendo o único programa que permite manipular folhas de cálculo. A adaptação do Excel às necessidades de cada utilizador não é propriamente novidade. As áreas que mais usam o Excel são as áreas financeiras e estatísticas. A área da engenharia normalmente utiliza outro tipo de ferramentas cujo desempenho é superior ao do Excel. No entanto, uma vez que o Excel é um produto que está muito difundido na sociedade e sua licença de utilização é mais barata que qualquer outra ferramenta especializada, algumas empresas optam por utilizar o Excel. O desempenho pode por vezes sair penalizado mas os objectivos são igualmente cumpridos.

A área da biologia apresenta um grande desenvolvimento e deste modo são necessárias novas ferramentas informáticas para que estes profissionais possam rentabilizar o seu trabalho. Os investigadores por vezes já utilizam o Excel para os ajudar a organizar e analisar dados resultantes das suas pesquisas. Por vezes há algumas referências em "*blogs*" [7] ou mesmo páginas pessoais que já existem algumas funcionalidades a serem adicionadas ao Excel para permitir aos investigadores extrair informação de tabelas de dados. No entanto, não foi possível obter nenhum documento cientificamente validado para confirmar esta informação.

Estas são então as principais razões que motivaram a implementação de novas funcionalidades para o Excel. Assim, pode-se aproveitar a interface e as funcionalidades do Excel para organizar, manipular e extrair conhecimento científico no ramo da biologia.

Primeiramente, estudou-se *Visual Basic for Applications* (VBA) e a possibilidade da sua utilização para implementar a ferramenta através desta tecnologia. Esta é uma tecnologia baseada em Visual Basic 6 que muitos utilizadores de Excel usam para implementar rotinas em folhas de cálculo.

VBA é uma tecnologia mais antiga e por este motivo existe mais documentação e os desenvolvedores de aplicações para Office poderão ter tendência a continuar a usar VBA uma vez que já estão familiarizados com esta tecnologia. As aplicações construídas em

[7] http://savas.me/blog/617

VBA podem ser executadas em quase todos os computadores, uma vez que todas as versões após o Office 97 suportam esta tecnologia.

No entanto, estas de aplicações podem ter graves problemas de segurança. Muitas vezes, têm macros embebidas dentro do documento e não é possível determinar se estas macros são as originais ou foram alteradas por algum software malicioso [3]. Desta forma, ao utilizar VBA não se poderá garantir que as macros construídas nunca danificarão o sistema. Outro problema associado com a segurança é o facto dos utilizadores raramente terem privilégios de administração na utilização do seu computador de trabalho. As aplicações VBA por vezes exigem que o utilizador altere o nível de segurança do Excel, algo que o utilizador nem sempre terá permissões para fazer.

As aplicações VBA são construídas com base em código não manejável o que leva a quebras de desempenho quando é necessário interagir com código manejável. Para além disso, VBA não suporta herança, polimorfismo e interfaces. Assim podemos concluir que VBA não é a tecnologia mais indicada na construção de aplicações de grande dimensão. A forma como VBA lida com excepções e erros é também muito mais pobre em relação à forma como VSTO lida esses mesmos erros. VSTO usa o mecanismo existente em dotNET para lidar com excepções ou erros. Tipicamente, manipular erros em VBA consistia em ler um código de erro quando ocorria uma excepção.

Depois do aparecimento de VSTO, VBA tenderá a cair em desuso. Algumas funcionalidades de VBA, como por exemplo o acesso a WebServices foi suportado em versões anteriores do Office [16] e já não o é no Office 2007 [17]. Renato Haddad, elemento da equipa Microsoft, num artigo que apresenta a tecnologia VSTO finaliza com a seguinte frase: "*Bons estudos e lembre-se: No Stress, think .NET + VSTO!*" [18].

Visual Studio Tools for Office é uma tecnologia derivada da plataforma dotNET que permite manipular os vários programas do Office. Desta forma, permite a integração de muitos dos objectos nativos desta plataforma em folhas de cálculo do Excel ou desenvolver novos menus "*Ribbon*" para esta aplicação. É uma tecnologia que apesar de ser novidade, permite integrar muitas e novas funcionalidades para o Excel de tal forma que o Excel poderá ser confundido com uma aplicação Windows. Por outro lado, esta tecnologia encontra-se em contínuo desenvolvimento e permite a integração de WebServices. Para

além disso, a utilização desta tecnologia não impede o desenvolvimento de rotinas em VBA. Os utilizadores poderão construir tais rotinas para potencializarem o seu trabalho. Desta forma, a escolha desta tecnologia para desenvolver as ferramentas de integração de dados não foi uma decisão difícil uma vez que se trata da melhor tecnologia da Microsoft que permite adaptar as aplicações do produto Office às necessidades do utilizador.

A ferramenta foi projectada para ter uma estrutura como a que é representada na Figura 3.2.

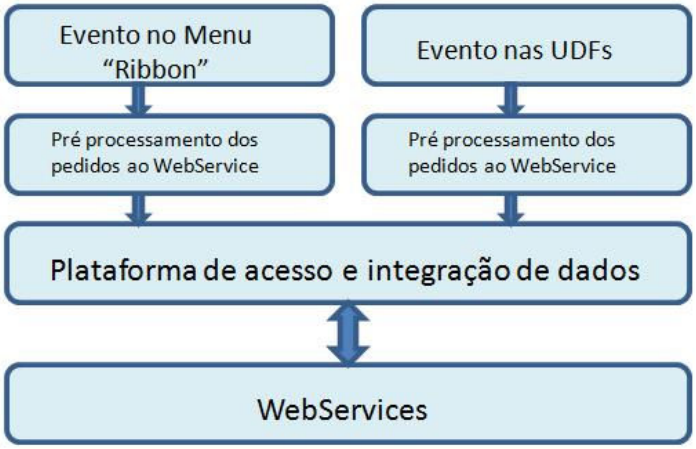


Figura 3.2 - Estrutura global da aplicação

Verificou-se que iriam existir grandes blocos de código que seriam executados tanto pelas User Defined Functions (UDF) como pelos botões do menu Ribbon. Desta forma, optou-se por estruturar a ferramenta com base em plataformas para permitir que grandes blocos de código fossem reutilizados. As UDF e os botões do menu "*Ribbon*" reagem a um evento e transferem a informação necessária para a plataforma de acesso a dados. Esta camada é responsável por realizar o consumo dos WebServices, organizar os dados devolvidos e integra-los no Excel.

Para além da análise das tecnologias que podem ser utilizadas para manipular folhas de cálculo, o projecto foi desenvolvido em duas fases:

- Construção de uma plataforma de acesso e integração de dados

- Construção de uma interface intuitiva e um conjunto de ferramentas de fácil utilização respeitando ao máximo a interface de utilização do Excel.

3.1 Acesso e integração de dados

3.1.1 WebServices

A integração dos dados em Excel representa um dos pontos cruciais para concretizar o objectivo pré-estabelecido. A integração dos dados para o Excel não pressupõe que os utilizadores tenham qualquer conhecimento da base de dados. Desta forma, foi construído um WebService com vários métodos que possibilitassem o acesso a diferentes tipos de dados. Os métodos também podem ser utilizados com vários parâmetros de entrada aumentando assim a diversidade dos dados que se podem extrair da base de dados.

A escolha de WebServices para aceder aos dados deveu-se ao facto destes poderem ser consumidos a partir de qualquer ponto do mundo e ao mesmo tempo estes utilizarem o protocolo HTTP para transmitir dados. Ao utilizarem este protocolo, fica desde logo assegurado que serão mínimos os problemas com as configurações de redes ou mesmo de configurações de *firewalls* uma vez que, maioritariamente, estas não impedem o acesso à rede Web. Para além disso, o GeNS já disponibilizava uma camada de WebServices. Desta forma, garante-se que a aplicação poderá ser utilizada em qualquer computador que tenha acesso à rede Web, tenha instalado o Microsoft Office 2007, a plataforma dotNET 3.5 e o "*Visual Studio Tools for Office Runtime 3.0*".

Os utilizadores dispõem de vários métodos para interagir com a base de dados. Existem métodos que permitem obter informação de organismos, proteínas, tipos de dados existentes na base de dados ou entidades biológicas. Para além destes, foram construídos métodos que permitem através de uma proteína saber os vários processos biológicos em que esta está envolvida ou vice-versa. Os novos métodos serão disponibilizados para uso público como pertencendo ao sistema GeNS. A Figura 3.3 apresenta um diagrama da estrutura da base de dados do GeNS.

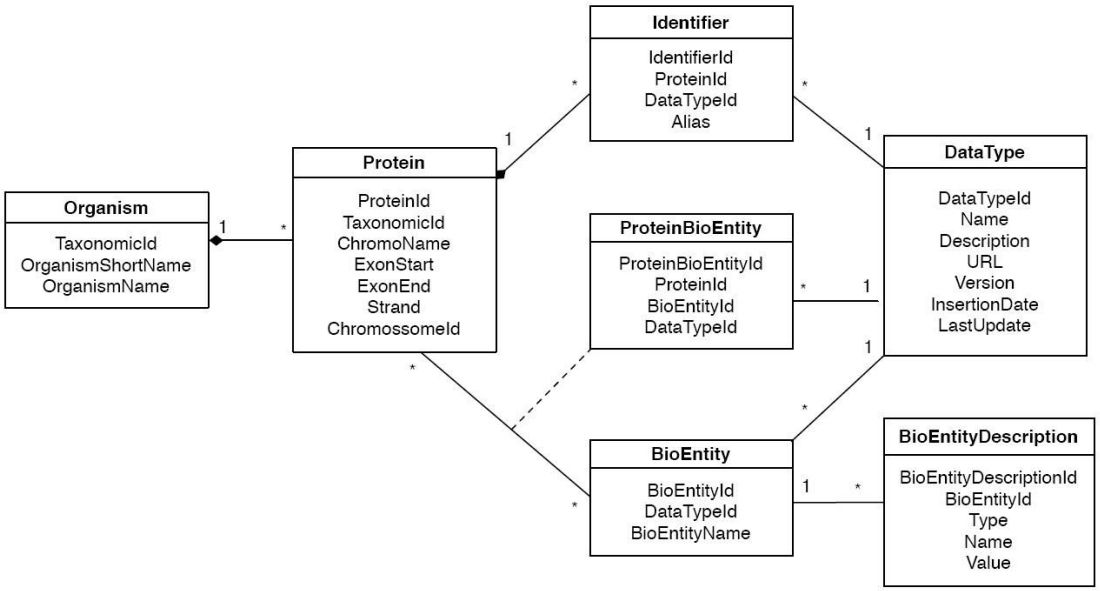


Figura 3.3 - Diagrama da base de dados do GeNS[8]

A Figura 3.4 apresenta uma lista dos métodos disponibilizados. Os métodos assinados na zona "V1" já existiam e os na zona "V2" foram implementados para permitir extrair mais informação da base de dados.

Figura 3.4 – Métodos Web existentes e desenvolvidos

[8] Diagrama gentilmente cedido por João Pereira, elemento do grupo de bioinformática da Universidade de Aveiro.

A Tabela 1 apresenta o tipo de informação que se poderá obter através do consumo de cada um dos métodos disponibilizado pelo WebService.

Tabela 1 - Descrição dos métodos do WebService do GeNS

	Método	Tipo de dados devolvidos
V1	SearchOrganism	Informação relativa a organismo (identificação taxonómica, nome curto, nome completo)
	ListDataTypes	Informação sobre os tipos de dados que existem na base de dados.
	SearchProtein	Diversa informação relativa a proteínas (e.g. nome do gene, nome proteína, "accession")
	SearchBioEntity	Informação relativa a entidades biológicas (e.g. KEGG pathway, gene ontology)
	ConvertProteinIdentifier	Possibilita obter uma correspondência directa entre quaisquer identificadores de proteínas
V2	GetProteinInfo	Informação relacionada sobre proteínas, e.g., relacionar genes com proteínas e vice-versa.
	GetProteinsInBioEntity	Informação relacionada entre proteínas e entidades biológicas, e.g., obter todas as proteínas que participam em determinado processo biológico
	GetBioEntityByProtein	Informação relacionada entre proteínas e entidades biológicas, e.g., obter as entidades biológicas onde determinada proteína está envolvida

Dos métodos apresentados na Tabela 1 apenas o método "*ConvertProteinIdentifier*" não se encontra integrado na ferramenta desenvolvida.

Os métodos do WebService permitem que sejam utilizados diferentes parâmetros de entrada e devolvem diferentes resultados mediante os parâmetros que lhes são passados. A partir destes parâmetros de entrada os utilizadores conseguem adaptar as pesquisas às suas necessidades. Para além dos parâmetros de pesquisa, no caso do método "*SearchProtein*" é também possível definir a quantidade ou o intervalo[9] de proteínas que se pretende incluir.

Os métodos pertencentes ao grupo V2 fizeram parte do desenvolvimento deste trabalho e desta forma será feita uma análise mais detalhada.

[9] O método SearchProtein permite que o utilizador escolha a gama de proteínas que pretende incluir. Por exemplo, as 10 primeiras ou as que se encontram entre a posição 10 e 20 na base de dados.

O método GetProteinInfo foi implementado para, por exemplo, permitir que a partir de um gene se consigam obter as proteínas que ele produz ou criar a relação inversa, i.e.,partindo do nome de uma proteína obter o gene que a produz. A Figura 3.23 presente na página 70 apresenta o resultado de uma execução desta função. Este método utiliza como parâmetros de entrada a identificação taxonómica, o tipo de dados actual, os dados actuais (e.g. nome do gene) e o tipo de dados que se pretende obter e devolve uma lista do nome das variáveis obtidas. O método SearchProtein devolve o campo "Alias" da tabela "Identifier". A partir deste campo e do correspondente tipo de dados, é obtido o "ProteinId" correspondente. Com a lista destes "ProteinIds" obtêm-se os campos "Alias" da tabela "Identifier" em que o "ProteinId" pertence à lista anterior e o tipo de dados é igual ao tipo de dados definido nos parâmetros de entrada dos WebServices (Figura 3.5).

```
gens.SelectCommand = "SELECT ProteinId into dbo.tmp" + conflux +
                     " FROM Identifier WHERE ((Alias= '" + proteinsAlias +
                     "') AND(DataTypeId=" + inputDataType + "))";
#region
gens.SelectCommand = "SELECT Alias FROM Identifier WHERE (ProteinId in "+
                        "(Select * from dbo.tmp" + conflux + ") AND DataTypeId=" +
                        outputDataType + ")";
```

Figura 3.5 - Código utilizado para obter informação sobre proteínas incluídas no GeNS

O método GetProteinsInBioEntity permite obter as proteínas que estão presentes em determinado processo biológico. Para utilizar este método é necessário conhecer a identificação taxonómica, o tipo de dados da entidade biológica e o nome da entidade biológica. Primeiramente, é obtido o "BioEntityId" a partir da tabela "BioEntity". Depois de obtida a identificação da entidade biológica é possível obter uma lista com as ProteinIds através da tabela "ProteinBioEntity". Finalmente, a partir da tabela Identifier, determina-se o tipo de dados (DataTypeId) e o campo "Alias" (Figura 3.6). Este método também poderá ser utilizado com um quarto parâmetro de entrada que será o tipo de dados a ser devolvido. Assim o utilizador poderá receber apenas informação sobre um determinado tipo de dados das proteínas que estão envolvidas em determinado processo biológico.

```
gens.SelectCommand = "SELECT BioEntityId into dbo.tmp"+conflux+
    " FROM BioEntity WHERE ((BioEntityName= '" + bioName+"') AND (DataTypeId=" + dataType + "))";
#region
gens.SelectCommand = "SELECT ProteinId into dbo.tmp" + conflux2 +
                     " FROM ProteinBioEntity WHERE (ProteinId in "+
                     "(SELECT ProteinID FROM Protein WHERE TaxonomicId = " + taxId.ToString()
                     + ")) AND (BioEntityId in (Select * from dbo.tmp"+conflux+"))";
#region
gens.SelectCommand = "SELECT Alias,DataTypeId FROM Identifier WHERE"+
                    " ProteinId in (Select * from dbo.tmp"+conflux2+")";
```

Figura 3.6 – Código utilizado para obter proteínas envolvidas numa entidade biológica

Foi ainda desenvolvido o método GetBioEntityByProtein. Para utilizar este método ter-se-á de conhecer a identificação taxonómica, o tipo de dados referente ao campo "Alias" e o campo "Alias" devolvido pelo método SearchProtein. Existe também a possibilidade de não definir a identificação taxonómica e assim esta informação será obtida para vários organismos. Os dados devolvidos por este método resultam de três pesquisas na base de dados. Primeiramente é obtido o "ProteinId" a partir da tabela "Identifier". Na segunda pesquisa, através da tabela "ProteinBioEntity", é obtida uma lista dos "BioEntityIds" em que o "ProteinId" pertence à primeira lista criada. Finalmente, o utilizador recebe os valores dos campos "BioEntityName" e "DataTypeId" da tabela "BioEntity".

```
gens.SelectCommand = "SELECT BioEntityId into dbo.tmp" + conflux +
                     " FROM ProteinBioEntity WHERE ProteinID in ^"+
                     "(Select ProteinId from Identifier WHERE Alias='" + proteinAlias+"')";
#region
gens.SelectCommand = "SELECT TaxonomicID into dbo.tmp" + conflux2 +
                     " FROM Protein WHERE ProteinID in (Select ProteinId"+
                     " from Identifier WHERE Alias='" + proteinAlias + "')";
#region
gens.SelectCommand = "SELECT BioEntityName,DataTypeId FROM BioEntity "+
                    "WHERE BioEntityId in (SELECT * from dbo.tmp" + conflux + ") ";
```

Figura 3.7 – Código utilizado para obter informação sobre as entidades biológicas em que determinada proteína está envolvida

A ferramenta apresentada neste documento contém para cada um dos métodos aqui apresentados um botão no menu "Ribbon" e uma "User Defined Function". A interface será detalhadamente apresentada neste capítulo.

3.1.2 Plataforma de acesso e integração de dados

Numa fase inicial, foi implementada uma forma que permitisse consumir os WebServices e importar os dados directamente para o Excel. Os dados obtidos eram automaticamente importados em folhas de cálculo. A Figura 3.8 representa de uma forma esquemática a forma como se realizava o acesso aos dados.

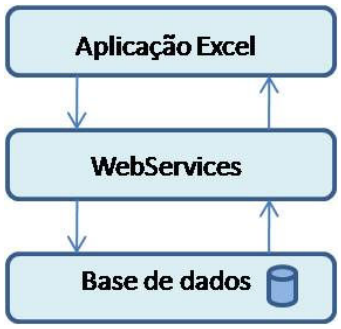


Figura 3.8 - Modelo de acesso a dados da aplicação

A tecnologia Visual Studio Tools for Office (VSTO) disponibiliza várias formas de integrar ou ler os dados em folhas de cálculo Excel e desta forma foram construídos métodos para permitir quer a leitura quer a escrita de valores nas células das folhas de cálculo. Por vezes surgiu a necessidade de incluir um grande volume de dados para folhas de cálculo. Uma vez que os métodos de integração de dados eram baseados na escrita de valores "célula a célula", o tempo necessário para o preenchimento de todas as células era muito elevado. Desta forma, a integração de dados em Excel não podia ser executada directamente uma vez que a aplicação iria ficar extremamente lenta[10].

Verificou-se que, caso se optasse por importar primeiramente os dados para um ficheiro de texto (formatado com uma estrutura tabular) e posteriormente importá-lo usando a ferramenta de importação de ficheiros de texto disponibilizada pela tecnologia VSTO, o tempo de integração dos dados diminuía drasticamente. A Figura 3.9 representa um

[10] Na secção de resultados é apresentado o resultado um teste aos três métodos desenvolvidos para integrar dados.

esquema da forma como os dados provenientes do WebService passaram a ser integrados no Excel.

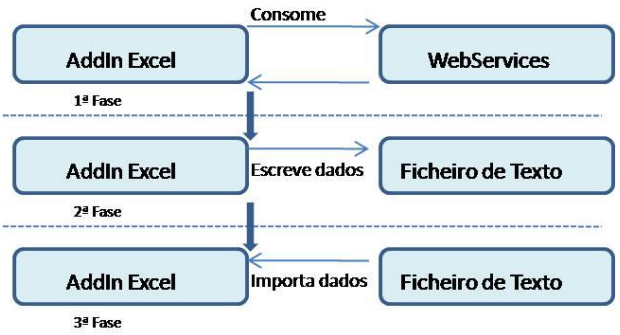


Figura 3.9 - Modelo do processamento dos dados retornados pelo WebService

A integração destes dados também era realizada de uma forma transparente para o utilizador.

Para utilizar este modelo de acesso e integração de dados seria necessário guardar ficheiros de texto em disco rígido. Desta forma, houve a necessidade de garantir que o utilizador possuía permissões de escrita de ficheiros. Primeiramente, pensou-se em obrigar o utilizador a gravar o livro de trabalho para detectar um caminho físico onde fosse possível escrever dados. Após alguma pesquisa, verificou-se que a plataforma dotNET é capaz de gravar ficheiros num caminho físico mesmo que o utilizador não tenha permissões de escrita. Assim, optou-se por utilizar esta funcionalidade da plataforma dotNET para gravar temporariamente os ficheiros de texto. Esta funcionalidade chama-se "*Isolated Storage*". Desta forma garante-se que por menos privilégios que o utilizador tenha, vai sempre ser possível gravar ficheiros no disco rígido.

Através deste segundo modelo de integração de dados, era possível garantir que o tempo de resposta da ferramenta construída não iria ser excessivamente elevado. No entanto, esta solução de escrever os dados em ficheiros de texto e posteriormente importa-los para Excel não foi uma decisão unânime. Seria no mínimo estranho que VSTO, uma tecnologia de manipulação de uma aplicação que está desenhada para lidar com estruturas tabulares, não permitisse integrar ou ler uma matriz ("Array") de dados do Excel de uma forma simples e rápida. Após a solução de integrar dados a partir de um ficheiro de texto já estar

implementada, verificou-se que existe a possibilidade de integrar um vector de dados ou uma matriz de dados directamente para Excel. O método que permite realizar esta tarefa pertence aos métodos disponíveis para manipular folhas de cálculo e tem por nome "*get_range( )*". Este método requer dois parâmetros de entrada que correspondem às células limite da zona que se pretende manipular, e.g., "*get_range(A1, F1)*" permite manipular as células contidas no intervalo entre a célula A1e a célula F1.

A partir da análise da Tabela 4 presente na página 64, verifica-se que este último método apresentado é o que permite integrar dados mais rapidamente em folhas de cálculo, independentemente do volume de dados. Por outro lado, apesar de ter sido desenvolvida uma forma de mostrar que apresenta progresso no acesso e integração dos dados (Figura 3.24, página 73 canto inferior esquerdo da imagem), os dados resultantes do consumo de WebServices vão sendo apresentados nas folhas de cálculo o que por si só é um elemento que mostra que a aplicação encontra-se em processo de acesso ou integração de dados. Para além disso, garante-se que mesmo que os métodos consumidos retornem diferentes quantidades de dados, a aplicação importa para o Excel todos os dados devolvidos uma vez que o vector de selecção das células que serão afectas é criado dinamicamente e tem em consideração os resultados devolvidos pelos WebServices. É importante de referir que o Excel 2007 tem limite máximo de linhas (1 048 576) e colunas (16 384) e que desta forma será emitido um aviso ao utilizador caso tente exceder estes limites.

Assim, o modelo de acesso a dados que se encontra implementado é o modelo apresentado na Figura 3.8 mas utiliza o método "matricial"[11] para integrar os dados. No entanto, caso os utilizadores estejam a tentar consumir WebServices não pertencentes ao GeNS os dados resultantes serão sempre escritos num ficheiro XML e desta forma será utilizado o segundo modelo representado na Figura 3.9. A ferramenta de importação de XML disponibilizada pela tecnologia VSTO utiliza como parâmetro de entrada o endereço físico do local onde se encontra o ficheiro XML a importar e é este o motivo pelo qual é utilizado o segundo método de acesso e integração de dados.

[11] O método "*get_range()*" permite seleccionar uma matriz de células do Excel e atribuir valores a cada entrada da matriz.

A Figura 3.10 apresenta o resultado da importação de XML do método "*SearchProteins (9606,1,5)* " disponibilizado pelo sistema GeNS[12].

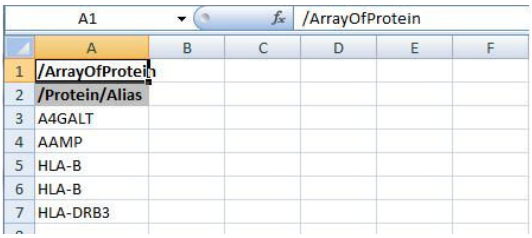


Figura 3.10 - Exemplo de importação XML de um WebService

A forma como os dados foram consumidos para apresentar a Figura 3.10 é substancialmente diferente da forma normal de consumo de WebServices. Para criar esta figura, não foi adicionada a referência aos WebServices. Foi feito, por via programática, um pedido HTTP e esperou-se pela resposta a este pedido. Uma vez que se estavam a consumir WebServices, a resposta é XML. O XML foi escrito num ficheiro com a mesma extensão. De seguida, utilizou-se o método da tecnologia VSTO que permite importar XML para uma nova folha de cálculo.

3.2 Interface

Na construção da interface havia vários desafios a cumprir. O principal objectivo da aplicação era integrar novas funcionalidades ao Excel e utilizar a sua interface para incluir dados provenientes de WebServices nas folhas de cálculo. Assim, era essencial que a ferramenta a construir respeitasse ao máximo o modo de utilização do Excel.

Desta forma, foram construídas duas formas de acesso aos dados:

- Através de botões num novo menu "*Ribbon*"
- Através de "*User Defined Functions*" (UDF).

[12] Foi utilizado "http://bioinformatics.ua.pt/GeNS/WS/Service.asmx/SearchProtein?data=9606,1,5" como parâmetro de entrada.

As duas interfaces construídas partilham algumas funções e é possível realizar as mesmas operações através de qualquer das interfaces. É importante referir que os utilizadores necessitam de um conhecimento do funcionamento global dos WebServices e das funções que estes disponibilizam para utilizar esta ferramenta. Os botões presentes no menu possuem ajuda e descrevem o modo da sua utilização. No entanto não fazem uma descrição exaustiva dos serviços disponibilizados e do tipo de parâmetros de entrada e saída destes mesmos serviços. Assim, assumir-se-á que os utilizadores leram atentamente o documento que explica o funcionamento e os métodos dos WebServices[13] e respeitam as regras da sua utilização. Nas funções construídas, foram testadas algumas condições para proceder à verificação dos parâmetros de entrada. No entanto, diferentes métodos poderão ser utilizados com os mesmos parâmetros de entrada e desta forma não é possível verificar se os dados obtidos são os que o utilizador pretendia obter. Desta forma, o desconhecimento dos WebServices poderá provocar erros ao analisar e extrair conhecimento dos dados incorporados nas folhas de cálculo.

3.2.1 Menu "Ribbon"

No novo menu construído, Excel Bio Tools (Figura 3.11), existem dois grupos principais:

- "*Where to include data*" (grupo da esquerda)
- "*Excel GeNS Tools*" (grupo da direita).

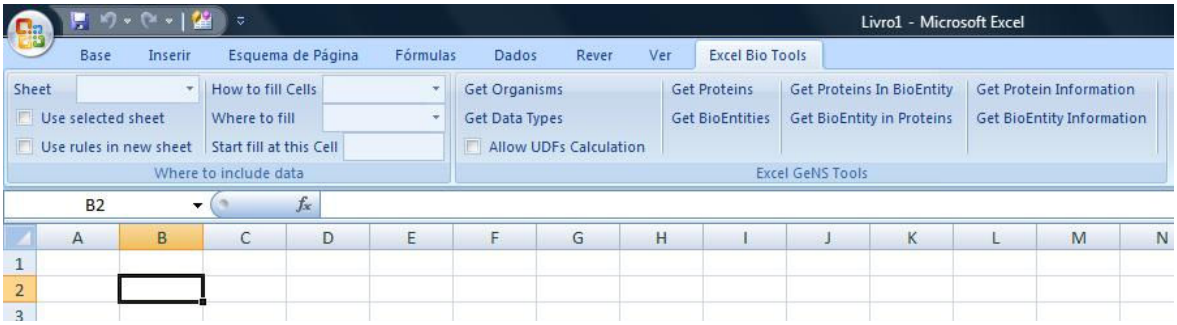


Figura 3.11 - Menu Ribbon construído

[13] Este documento poderá ser consultado em http://bioinformatics.ua.pt/Members/pedrolopes/GeNS%20-%20Web%20Services%20Description.pdf

O grupo "*Excel GeNS Tools*" possui o conjunto de funções disponibilizadas para integrar dados no Excel. No grupo "*Where to include data*" os utilizadores poderão escolher a forma como querem organizar os dados que são importados para o Excel.

No grupo "*Where to include data*", os utilizadores poderão escolher se preferem que os dados sejam incluídos numa nova folha de cálculo ou numa folha já existente. Para além disso, poderão organizar os dados numa coluna, numa linha ou apenas numa célula. Este grupo é comum à utilização dos botões disponíveis nos menus ou ao uso de UDF. As opções definidas neste grupo serão utilizadas pelas funções implementadas na plataforma de acesso e integração de dados.

Para utilizar os botões do grupo "*Excel GeNS Tools*" os utilizadores deverão introduzir nas células da folha de cálculo os parâmetros das pesquisas que pretendem que sejam efectuadas. As UDF podem ser utilizadas para ajudar os utilizadores a construir os parâmetros de acesso aos WebServices. No entanto, o uso destas UDF não é de carácter obrigatório. Desta forma, caso os utilizadores se sintam suficientemente familiarizados com a ferramenta os parâmetros de entrada dos WebServices podem ser introduzidos directamente nas células. Posteriormente os utilizadores deverão seleccionar as células que contêm os estes parâmetros e clicar no botão correspondente à função que desejam executar (Figura 3.12).

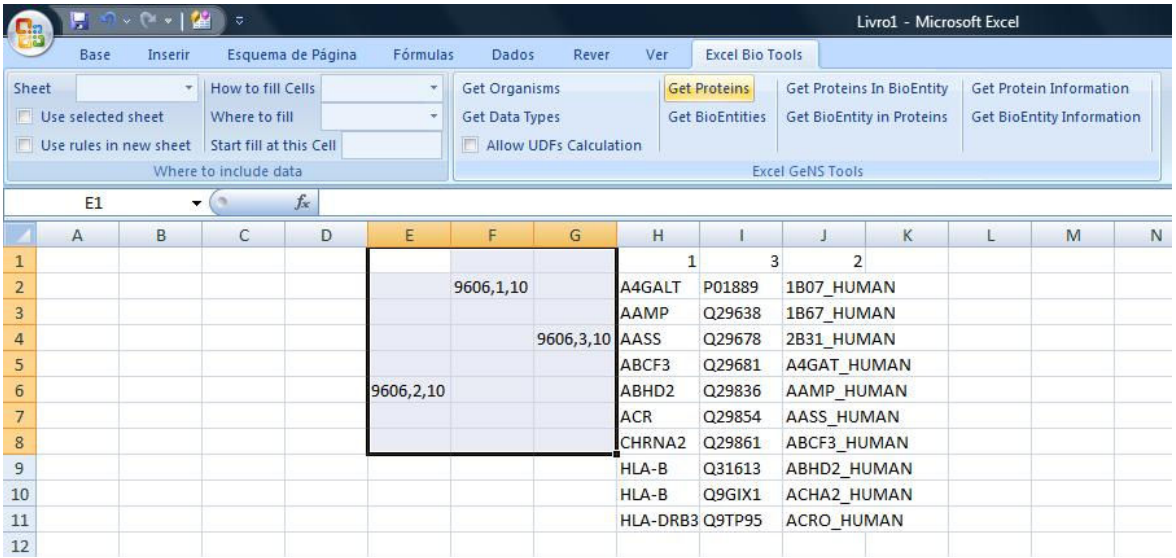


Figura 3.12 - Exemplo de utilização da interface desenvolvida no menu Ribbon

Após o clique, o programa irá começar a execução. A aplicação construída irá aceder aos WebServices e retornará os dados para o local definido pelo utilizador.

Apesar de existir uma triagem inicial na execução de cada função, o conjunto de células seleccionadas deverá apenas conter valores que devem ser calculados por esta função. Este ponto é de extrema importância uma vez que diferentes métodos poderão possuir os mesmos parâmetros de entrada mas conduzem a resultados completamente diferentes. O programa emite um aviso caso existam diferentes métodos na selecção utilizada, mas apenas se estes utilizarem as UDF construídas para construir os parâmetros de entrada dos WebServices. Caso tal não se verifique, o programa continuará a sua execução normal e os dados devolvidos serão introduzidos na folha de cálculo. Caso o utilizador não defina como quer organizar os dados, os botões do grupo "*Excel GeNS Tools*" irão introduzir os dados resultantes do consumo dos WebServices na primeira coluna não utilizada da folha de cálculo que se encontra activa (Figura 3.12).

Para construir esta interface, foi usado e Visual Studio 2008 Professional. Este programa possui todas as ferramentas necessárias e modelos com os vários objectos que podem ser utilizados para implementar uma interface num novo menu.

Na construção desta interface foram adicionados botões, caixas de texto, lista "drop-down" e "checkboxs". O Visual Studio já tem modelos para estes componentes e utilizando a visão de desenho é apenas necessário arrastá-los para o novo menu. Posteriormente, ter-se-ão de manipular os eventos necessários para que estes controlos reajam às acções do utilizador e executem o código que permitirá realizar as operações desejadas. Para o caso desta interface, apenas foram manipulados os eventos de clique. Após o clique, o programa irá verificar as células que se encontram seleccionadas e será criada uma lista com os valores que corresponderem a parâmetros que podem ser utilizados para consumir os WebServices. Caso haja células que contenham valores que não respeitem a utilização dos WebServices, estes valores serão descartados. A lista de pedidos de consumo de WebServices será enviada para a plataforma de acesso e integração de dados. Esta plataforma é responsável por consumir os WebServices e retornar os dados para o Excel.

A Tabela 2 apresenta uma lista dos botões e métodos dos WebServices que cada um dos botões executa.

Tabela 2 - Correspondência entre botões e os métodos dos WebServices

Botão	Método
Get organisms	SearchOrganism
Get data types	ListDataTypes
Get proteins	SearchProtein
Get bioentities	SearchBioEntity
Get proteins in bio entity	GetProteinsInBioEntity
Get bioentity in proteins	GetBioEntityByProtein
Get protein information	GetProteinInfo

3.2.2 User Defined Functions

Características das UDF

"*User Defined Functions*" são funções que têm como objectivo expandir as funcionalidades do Excel. Estas novas funções podem ser desenvolvidas usando código manejável ou código não manejável. Caso sejam desenvolvidas usando a plataforma dotNET, e consequentemente desenvolvidas em código manejável, são criados *"COM Callable Wrappers (CCWs) "* para permitir que os objectos COM possam interagir com o código manejável desenvolvido em dotNET.

Como foi referido anteriormente, é possível criar novas funções definidas pelo utilizador ("*User Defined Functions"* – UDF) que podem ser utilizadas tal como as fórmulas já existentes no Excel. Para tal, o programador terá de criar um ficheiro do tipo XLL [19], [20]. Este tipo de ficheiros pode ser visto como um tipo especial de DLL. Para desenvolver este tipo de funções poder-se-á utilizar VBA. Dependendo do contexto de utilização e da opção de cada utilizador, as UDF irão permitir aos utilizadores criar facilmente os parâmetros de entrada que irão ser utilizados no consumo de WebServices ou chamar directamente os WebServices. Para tal, os utilizadores deverão colocar ou retirar o visto da caixa presente no grupo "*Excel GeNS Tools*" para permitir que as UDF efectuem ou não chamadas a WebServices. Este controlo apenas existe para permitir que os botões do menu

possam ser executados sem duplicar o acesso aos WebServices. Caso as UDF efectuassem sempre o acesso aos WebServices estas não poderiam ser utilizadas como meros auxiliares à construção dos parâmetros de entrada dos WebServices.

Também existe a possibilidade de construir estas funções utilizando C# ou VB.NET sem que seja necessário recorrer a XLLs. Para tal, ter-se-á de construir uma nova DLL. Esta tecnologia é conhecida como "*automation add-in*" [21]. A Figura 3.13 apresenta uma das funções construídas. Após programar as funções desejadas é necessário, nas opções de construção do projecto no Visual Studio, indicar que esta nova aplicação deverá ser registada como tendo interoperabilidade com COM. Deste modo, as classes definidas como tendo interoperabilidade com COM, estarão disponíveis para serem utilizadas no Excel.

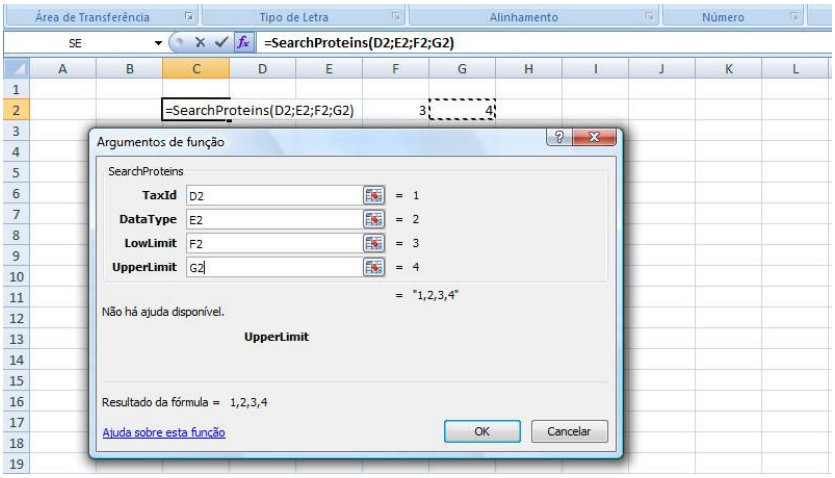


Figura 3.13 - Exemplo de uma "*User Defined Function*"

A Figura 3.14 apresenta de uma forma mais detalhada a forma como componentes dotNET podem utilizar componentes COM e os componentes COM podem utilizar componentes dotNET.

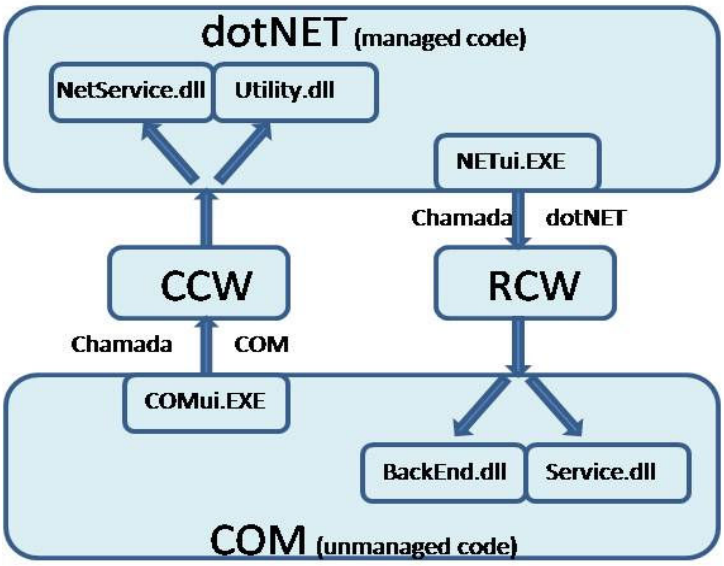


Figura 3.14 - Diagrama que representa a chamada entre a plataforma dotNET e componentes COM [23], [24]

Na Figura 3.14 pode-se observar que existe como que um proxy que é responsável por fazer a ponte entre código manejável e código não manejável.

As UDF apresentam algumas limitações e uma delas é que estas apenas têm a capacidade de apenas alterar o valor da célula em que estão a ser calculadas [22]. Quaisquer tentativas de modificar valores noutras células serão automaticamente ignoradas pelo Excel. As UDF poderão invocar outros procedimentos. No entanto, estes têm as mesma limitações que as UDF. Por outro lado, no *"scope"* das UDF é possível consumir WebServices. No entanto, as funções desenvolvidas nestas UDF têm muitas restrições [25] e todo o tipo de dados incluídos dentro do seu *"scope"* têm de ser suportados por objectos COM [26]. Caso se tente declarar e inicializar um tipo de dados não nativo, o Excel apresentará "#VALOR" como resultado da execução da função sem retornar qualquer tipo de excepção [22].

Implementação das UDF

As funções do Excel são uma das principais funcionalidades deste software. Estas funções estão potencializadas para aplicar uma fórmula a um vasto conjunto de células. O resultado destas funções é colocado na mesma célula em que o utilizador introduziu os dados de entrada que permitem o cálculo da função. Por exemplo, se se colocar a função *"=SOMA*

(A1;B1) " na célula C1, o valor que será apresentado na célula C1 é o resultado da soma dos valores contidos nas células A1 e B1. Uma das primeiras funcionalidades que os utilizadores de Excel aprendem a utilizar é a capacidade de aplicar uma fórmula construída às células seguintes. Através de um simples clique e arrasto, é possível definir que a célula C2 contenha a função *"=SOMA (A2;B2) "*. Esta funcionalidade permite que a mesma fórmula seja aplicada a inúmeras células contidas na folha de cálculo sem que seja necessário um grande esforço por parte do utilizador.

Desta forma, desde o início deste projecto havia uma enorme vontade de construir UDF que permitissem aos utilizadores consumir WebServices. Na implementação destas UDF foram surgindo alguns problemas que apesar de já terem sido descritos neste documento, se achou relevante voltar a referi-los:

- Limitação nos tipos de dados que podem ser manipulados no "*scope*" UDF, o que inviabiliza o consumo de WebServices
- Impossibilidades de as UDF poderem alterar valores de outras células.

Estes dois pontos limitavam fortemente a capacidade das UDF e estas não poderiam realizar as operações básicas necessárias ao funcionamento a aplicação. Assim, decidiu-se construir UDF que permitissem construir os parâmetros de entrada dos WebServices. Aqui, também se pressupõe que os utilizadores têm um conhecimento mínimo das funções disponibilizadas pelos WebServices. Apesar das UDF não disponibilizarem ajuda, estas possuem uma interface gráfica muito intuitiva (Figura 3.13).

Na verdade, a única função que as UDF desempenham é a verificação dos parâmetros de entrada e caso tenham sido inseridos parâmetros que permitirão o consumo dos WebServices será construída a frase de acesso. Caso tal não se verifique, as funções retornarão a mensagem "*Invalid Usage*" na célula onde se tenta efectuar este cálculo. A Figura 3.13 presente na página 57 apresenta o exemplo de uma das funções construídas que permite construir a frase que permitirá o consumo dos WebServices.

Desta forma, as UDF poderão parecer quase inúteis. No entanto, estas funções poderão ser utilizadas nas duas interfaces construídas. Na primeira interface, já apresentada neste documento, deverão apenas servir como uma ferramenta auxiliar para construir os

parâmetros de entrada dos WebServices. No entanto, na segunda interface desempenham um papel fulcral.

Como foi anteriormente referido, VSTO é uma tecnologia pertencente à plataforma dotNET que permite construir novas funcionalidades para o Excel e manipular as folhas de cálculo existentes neste produto do Office. Esta tecnologia permite desenvolver métodos que reagem a eventos criados pelas acções dos utilizadores. São inúmeros os eventos que é possível monitorizar com VSTO. Existe um que é especialmente importante para este trabalho, o evento "*ActiveWorkbook_SheetCalculate*".

Como se pode perceber pelo próprio nome do evento, este ocorre sempre que é efectuado um cálculo numa das células do livro que se encontra aberto. Assim, caso se adicione a este evento lógica programática, é possível detectar se o utilizador pediu que fosse efectuado um cálculo que utiliza uma das UDF ou uma função já existente do Excel.

Uma vez que já não estamos a lidar com UDF todos os problemas relacionados com o desenvolvimento destas funções desaparecem. Torna-se então possível tirar partido das funcionalidades da plataforma dotNET e da tecnologia VSTO para desenvolver novas funcionalidades para o Excel. Para o utilizador, estas funções serão em tudo semelhantes às funções que o Excel já possui. No entanto, na óptica do programador as UDF apenas lhe fornecem informação se o evento deverá ou não executar a lógica necessária para satisfazer as necessidades dos utilizadores.

Para se conseguirem consumir os WebServices é necessário obter o valor dos parâmetros de entrada necessários à execução dos seus métodos. Estes valores estão contidos nas células em que as UDF efectuaram cálculos. Verificou-se quando se aplicava uma fórmula a uma única célula, poderia ser difícil detectar qual a célula que continha os parâmetros de entrada dos WebServices. Por este motivo, através do evento "*SheetSelectionChange*" é sempre guardado o valor da última célula activa. Este problema não existe quando se aplica a função de arrasto de fórmula às células vizinhas uma vez que o conjunto de células que ficam seleccionadas é sempre o grupo de células que contêm os parâmetros de entrada dos WebServices.

No evento "*ActiveWorkbook_SheetCalculate*", é feita uma triagem das funções que provocaram o evento. Caso se tratem de UDF pertencentes à lista das UDF construídas,

será criada uma lista com os parâmetros de entrada dos WebServices e os métodos que serão chamados. Tal como nas funções do menu Ribbon, esta lista com os métodos e parâmetros de entrada a serem utilizados será passada para a plataforma de acesso e integração de dados.

No grupo "*Where to include data*", foi disponibilizada a possibilidade de incluir os dados numa única célula, separados por ';' (Figura 3.15). Esta opção tenta aproximar ao máximo as UDF à forma como as funções Excel devolvem os dados. Por vezes surgirá a necessidade de colocar os dados em novas colunas, linhas ou noutras células. Neste caso, antes de utilizar as UDF os utilizadores deverão definir no menu "*Where to include data*" como desejam que os dados sejam introduzidos na folha de cálculo.

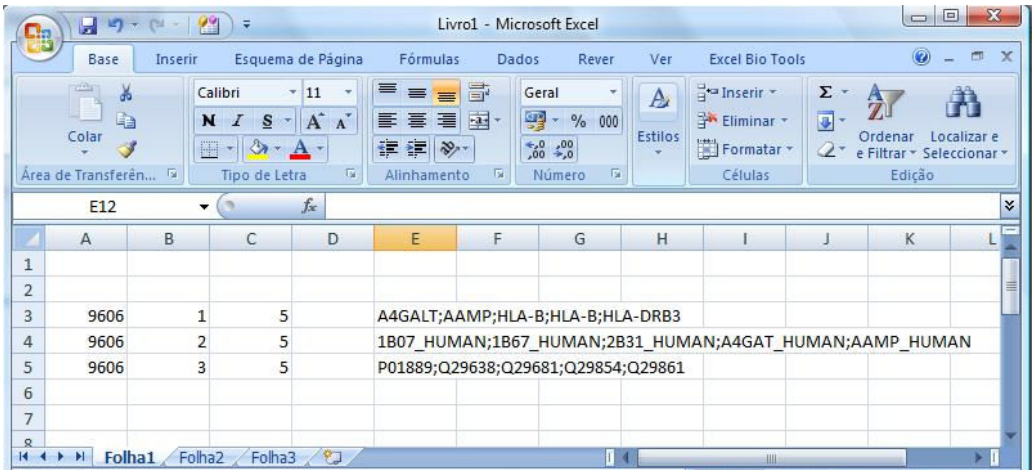


Figura 3.15 - Interface das UDF

O objecto "*range*" permite manipular uma célula ou um conjunto de células. No entanto, ao apresentar os valores na mesma célula que continha a fórmula perde-se a fórmula de cálculo e desta forma será impossível utilizar a função de arrasto de fórmula às células vizinhas. Assim, as UDF devem ser utilizadas, sempre que possível, de forma a retornar valores para células que não as que contêm a formula. Desta forma, para criar a Figura 3.15 tiveram que se escrever as fórmulas nas células E3, E4 e E5 isoladamente.

A tabela de correspondência entre UDF e métodos dos WebServices é muito semelhante à tabela de correspondência entre os métodos dos WebServices e os botões do menu Ribbon.

O nome das UDF por vezes é diferente do nome atribuído para os botões. No entanto, este nome ainda poderá vir a ser alterado para manter coerência entre UDF e nome dos botões do menu "Ribbon". A Tabela 3 apresenta a correspondência entre UDF e métodos dos WebServices.

Tabela 3 - Correspondência entre UDF e métodos dos WebServices

UDF	Método
SearchOrganism	SearchOrganism
ListDataTypes	ListDataTypes
SearchProteins	SearchProtein
SearchBioEntity	SearchBioEntity
GetProteinsInBioEntity	GetProteinsInBioEntity
SearchBioentityByProtein	GetBioEntityByProtein
GetProteinInformation	GetProteinInfo

A Figura 3.16 apresenta um diagrama de classes da aplicação construída.

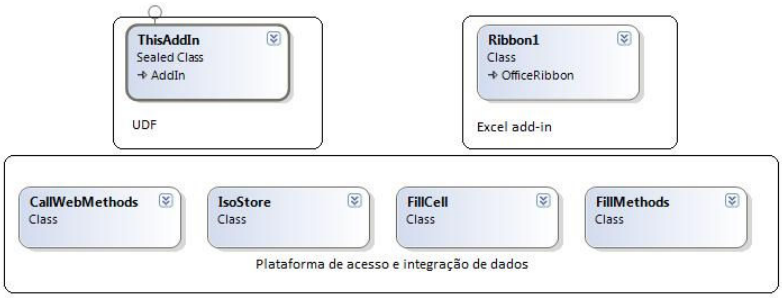


Figura 3.16 - Diagrama de classes

Na classe "*ThisAddIn*" estão definidos os manipuladores de eventos que permitem utilizar as User Defined Functions (UDF). Esta classe possui o evento "*ThisAddIn_Startup*" (Figura 3.17) que ocorre sempre que o Excel é iniciado. Desta forma, este evento é ideal para garantir que se monitorizam todos os eventos necessários para permitir que as UDF sejam utilizadas.

```
private void ThisAddIn_Startup(object sender, System.EventArgs e)
{
    startup definitions

    #region Event handlers

    Globals.ThisAddIn.Application.SheetSelectionChange +=
        new Microsoft.Office.Interop.Excel.AppEvents_SheetSelectionChangeEventHandler(Application_SheetSelectionChange);
    Globals.ThisAddIn.Application.AfterCalculate +=
        new Microsoft.Office.Interop.Excel.AppEvents_AfterCalculateEventHandler(Application_AfterCalculate);
    Globals.ThisAddIn.Application.ActiveWorkbook.NewSheet +=
        new Microsoft.Office.Interop.Excel.WorkbookEvents_NewSheetEventHandler(ActiveWorkbook_NewSheet);
    Globals.ThisAddIn.Application.ActiveWorkbook.SheetCalculate +=
        new Microsoft.Office.Interop.Excel.WorkbookEvents_SheetCalculateEventHandler(ActiveWorkbook_SheetCalculate);
    Globals.ThisAddIn.Application.SheetDeactivate +=
        new Microsoft.Office.Interop.Excel.AppEvents_SheetDeactivateEventHandler(Application_SheetDeactivate);

    #endregion
}
```

Figura 3.17 - Evento de "startup" do add-in

Na classe “Ribbon1” estão definidos os manipuladores de eventos de clique nos botões desenhados no menu “Ribbon”. Os eventos existentes nestas duas classes preparam os parâmetros de entrada que vão ser utilizados pela classe “CallWebMethods”. Esta classe é responsável por consumir os WebServices. Os métodos desta classe utilizam os métodos da classe “*FillMethods*” ou “*IsoStore*” para integrarem os dados em folhas de cálculo. A classe “*FillCell*” possuía os primeiros métodos desenvolvidos para integrar dados em folhas de cálculo. Os seus métodos serão pouco utilizados. Esta classe apenas é apresentada neste documento porque é será realizado um teste de comparação do tempo de execução das três formas implementadas para integrar dados em Excel.

A implementação dos novos métodos para o WebService do GeNS e das UDF foi executada em dois projectos independentes. Os novos métodos foram adicionados ao projecto já existente. Desta forma, as classes que contêm estes métodos não foram apresentadas no diagrama de classes da aplicação Excel. No caso das UDF foi utilizado o processo conhecido como “automation add-in”.

3.3 Resultados

Testes de integração de dados

Foi realizado um teste para determinar o método mais rápido para integrar dados em Excel. Este teste consistiu em determinar o tempo necessário a integrar dados em Excel utilizando os três métodos desenvolvidos. A Tabela 4 apresenta os resultados obtidos.

Tabela 4 – Tempo, em milissegundos, de escrita de células em Excel, utilizando os diferentes métodos

Quantidade de dados	Matriz dados	Ficheiro texto	Célula a célula
10	5	408	28
100	6	413	180
1000	21	583	1536
10000	116	2061	28465
100000	679	853	Não testado
250000	1677	1942	Não testado
500000	3329	4805	Não testado

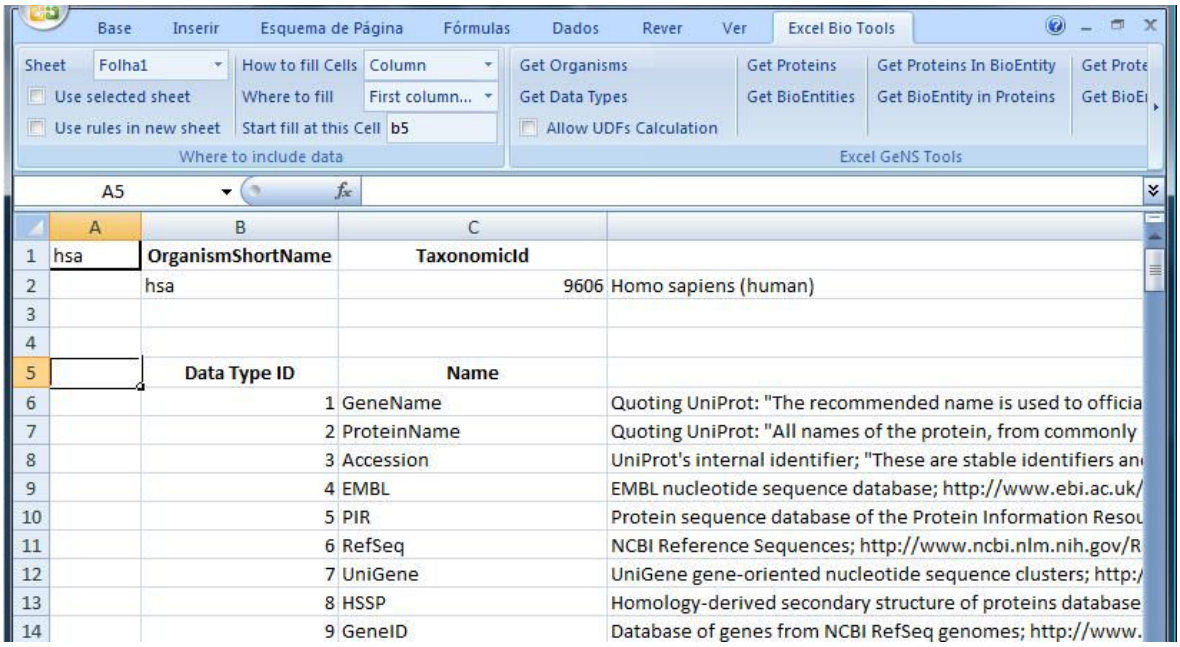
Para realizar este teste foi criado um vector com a quantidade de dados variável. O tempo necessário à criação deste vector não está incluído nos registos temporais apresentados na Tabela 4. Para cada uma das quantidades de dados, foram efectuados três registos temporais e foi encontrado o tempo médio. Para o preenchimento célula a célula, não foram testados os métodos que integram mais de dez mil dados uma vez que para este valor o tempo de resposta da aplicação já era demasiadamente lento.

Através da Tabela 4 pode-se verificar que tanto o método de integração de dados através de recurso a um ficheiro de texto como o uso de uma matriz podem ser utilizados para integrar dados para folhas de cálculo. Tipicamente, a integração de dados será realizada através do recurso a uma matriz de dados uma vez que este método é mais rápido.

Simulação de um possível cenário de utilização da ferramenta

Neste capítulo, foram previamente apresentados vários desafios que se foram levantando à medida que a interface com o utilizador foi desenvolvida. Como foi dito anteriormente, para além de um conjunto de WebServices, foram desenvolvidas duas interfaces para ajudar os investigadores a integrar no Excel os dados que irão ser alvo do seu estudo. Para melhor se perceber a relevância e as capacidades da ferramenta desenvolvida, será apresentado um exemplo de uma possível utilização da mesma. É de extrema importância voltar a referir que a utilização da ferramenta pressupõe que os utilizadores estão familiarizados com os WebServices.

Os investigadores têm por vezes a necessidade de obter dados sobre determinados organismos e posteriormente saber mais informação sobre esses organismos. Relativamente a estes organismos, a informação necessária é o nome completo do organismo, a sua identificação taxonómica e o seu nome curto. Depois de definido o organismo que será objecto de estudo, os investigadores poderão obter mais dados sobre este organismo, como por exemplo o nome dos genes, o nome das proteínas ou o identificador do gene. Assim, se os investigadores desejarem obter esta informação terão de conhecer o correspondente tipo de dados. Para obter a lista completa dos tipos de dados existentes na base de dados, os utilizadores deverão utilizar a correspondente UDF ou clicar no botão *"Get Data Types"*. A listagem dos tipos de dados permite que sejam obtidos apenas os tipos de dados que são relativos a proteínas ou dados relativos apenas a entidades biológicas. Estes dois passos (obter informação sobre os organismos e tipos de dados existentes na base de dados) são absolutamente necessários para que se possam realizar pesquisas sobre proteínas ou entidades biológicas. Para este exemplo, foi obtida informação sobre o ser humano e foram listados os tipos de dados existentes (Figura 3.18).



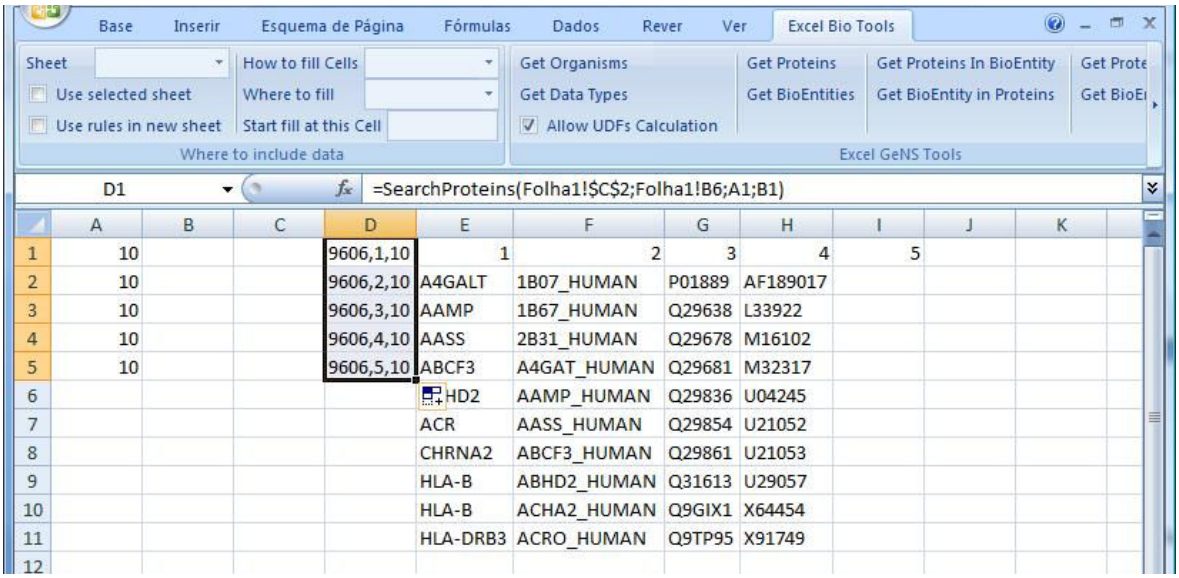
	A	B	C	
1	hsa	**OrganismShortName**	**TaxonomicId**	
2		hsa	9606	Homo sapiens (human)
3				
4				
5		**Data Type ID**	**Name**	
6		1	GeneName	Quoting UniProt: "The recommended name is used to officia
7		2	ProteinName	Quoting UniProt: "All names of the protein, from commonly
8		3	Accession	UniProt's internal identifier; "These are stable identifiers an
9		4	EMBL	EMBL nucleotide sequence database; http://www.ebi.ac.uk/
10		5	PIR	Protein sequence database of the Protein Information Resou
11		6	RefSeq	NCBI Reference Sequences; http://www.ncbi.nlm.nih.gov/R
12		7	UniGene	UniGene gene-oriented nucleotide sequence clusters; http:/
13		8	HSSP	Homology-derived secondary structure of proteins database
14		9	GeneID	Database of genes from NCBI RefSeq genomes; http://www.

Figura 3.18 - Integração de informação sobre organismos e tipos de dados

A partir destes dados os investigadores já poderão realizar pesquisas sobre outro tipo de informação. A maioria dos métodos disponibilizados pelo WebService necessitam da informação da identificação taxonómica do organismo e sobre o tipo de dados que se pretende obter e esta é a razão pela qual a obtenção deste tipo de informação é essencial.

O método "*SearchProtein*" do WebService permite ao utilizador obter informação sobre as proteínas de um determinado organismo. A informação que é obtida é sempre dependente dos parâmetros que o utilizador fornece. Deste modo, o utilizador poderá obter o nome dos genes, o nome das proteínas ou quaisquer outros tipos de dados. Para além disso, o utilizador poderá ainda definir o número de proteínas ou o intervalo[14] que pretende incluir. Para este exemplo, foi decidido que iriam ser integrados dados sobre o nome dos genes, o nome das proteínas, o "*Accession*", o "*EMBL*" e o "*PIR"* das mesmas para a folha de cálculo número dois (Figura 3.19).

[14] O método permite que o utilizador escolha a gama de proteínas que pretende incluir. Por exemplo, as 10 primeiras ou as que se encontram entre a posição 10 e 20 na base de dados.



	A	B	C	D	E	F	G	H	I	J	K
1	10			9606,1,10	1	2	3	4	5		
2	10			9606,2,10	A4GALT	1B07_HUMAN	P01889	AF189017			
3	10			9606,3,10	AAMP	1B67_HUMAN	Q29638	L33922			
4	10			9606,4,10	AASS	2B31_HUMAN	Q29678	M16102			
5	10			9606,5,10	ABCF3	A4GAT_HUMAN	Q29681	M32317			
6					HD2	AAMP_HUMAN	Q29836	U04245			
7					ACR	AASS_HUMAN	Q29854	U21052			
8					CHRNA2	ABCF3_HUMAN	Q29861	U21053			
9					HLA-B	ABHD2_HUMAN	Q31613	U29057			
10					HLA-B	ACHA2_HUMAN	Q9GIX1	X64454			
11					HLA-DRB3	ACRO_HUMAN	Q9TP95	X91749			
12											

Figura 3.19 - Integração de dados sobre proteínas

Para importar estes dados, foi utilizada uma UDF. Depois de definida a função na primeira célula, através do simples arrasto, a mesma fórmula é aplicada às células seguintes. Como se pode observar através da Figura 3.19, para além dos dados que resultam da execução da função, é também introduzido um cabeçalho com o número que corresponde ao tipo de dados que a coluna contém.

Para obter dados sobre as entidades biológicas existentes na base de dados poderá ser utilizada a mesma metodologia. Para este exemplo, para obter informação sobre entidades biológicas, foram seleccionadas as células que continham os parâmetros de entrada Figura 3.20. Os dados resultantes foram introduzidos para as colunas B e C. Mais uma vez, cada uma das colunas tem indicação sobre o tipo de dados que foi introduzido na folha de cálculo.

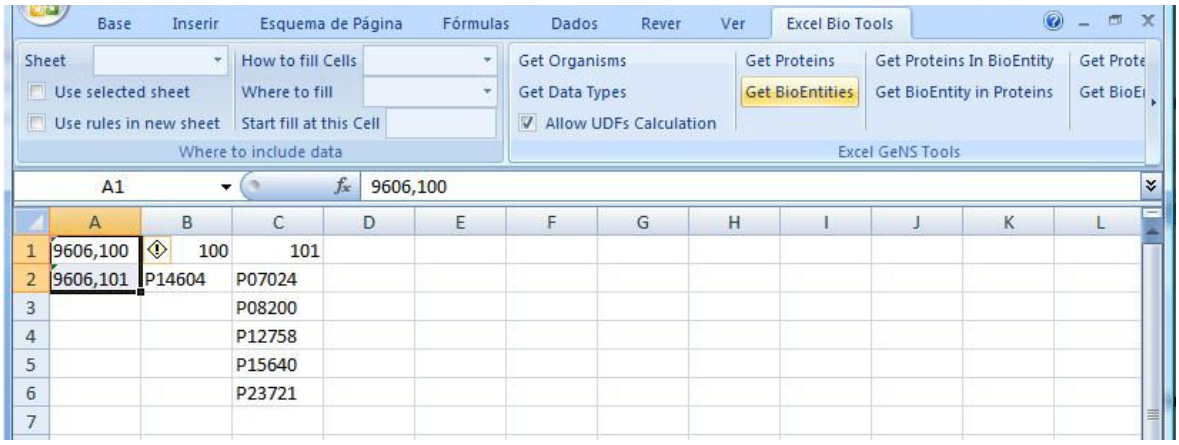
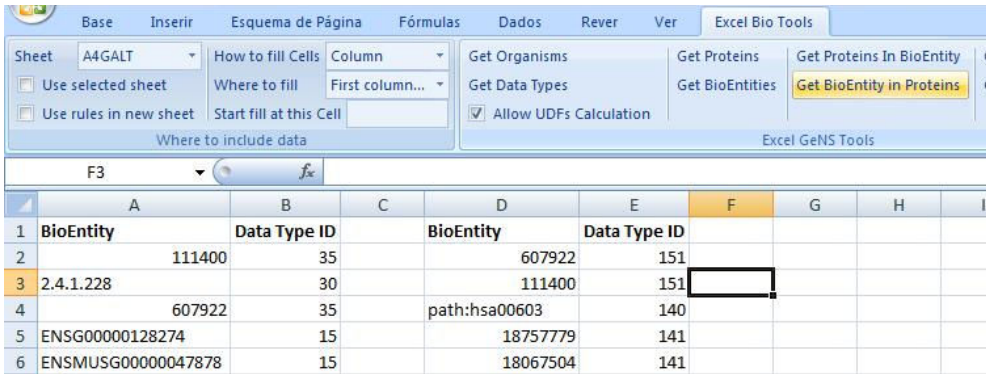


Figura 3.20 - Integração de dados sobre entidades biológicas

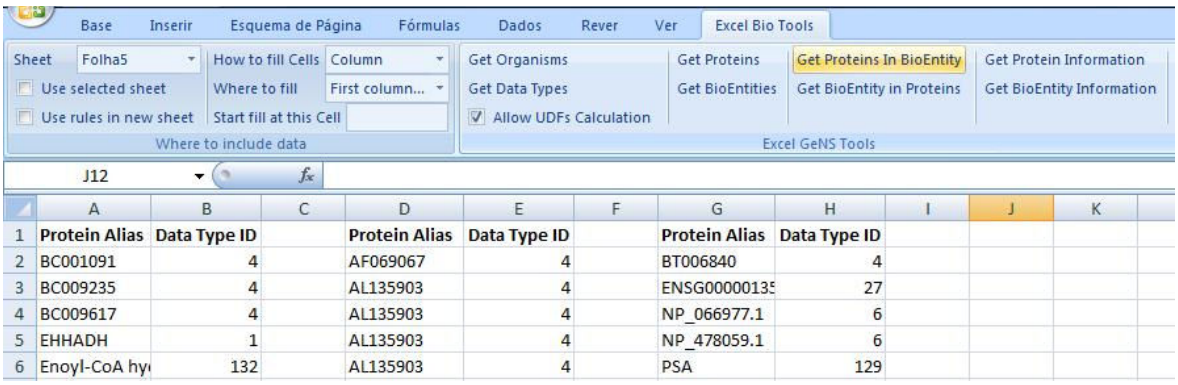
As funcionalidades até aqui apresentadas utilizam métodos que já estavam integrados no sistema GeNS. Este conjunto de métodos não permitia que fossem realizadas pesquisas que permitissem saber quais as proteínas que um gene produz ou quais as doenças que estão relacionadas com determinado gene. Por este motivo foram desenvolvidos mais métodos para permitir realizar este tipo de pesquisas. Assim, será possível relacionar genes com proteínas, proteínas com entidades biológicas ou vice-versa. A Figura 3.21 apresenta o resultado de uma destas pesquisas. Nesta pesquisa, encontram-se as entidades biológicas em que o gene "A4GALT" se encontra presente. Este método permite limitar a pesquisa por organismo, caso o utilizador assim o entenda. Os resultados apresentados, nas colunas A,B e D,E correspondem a dados para o mesmo gene. No entanto, nos resultados do lado esquerdo, foi feita uma pesquisa para determinar as entidades biológicas em que se pode encontra este gene sem depender do organismo. Nas colunas D e E estão apresentados os resultados mas limitando apenas a pesquisa para o ser humano. Através desta função será então possível identificar se determinado gene está presente em diversos organismos e quais as entidades biológicas em que ele está envolvido.



	A	B	C	D	E	F	G	H
1	**BioEntity**	**Data Type ID**		**BioEntity**	**Data Type ID**			
2	111400	35		607922	151			
3	2.4.1.228	30		111400	151			
4	607922	35		path:hsa00603	140			
5	ENSG00000128274	15		18757779	141			
6	ENSMUSG00000047878	15		18067504	141			

Figura 3.21 - Integração de dados sobre as entidades biológicas que determinada proteína está envolvida

Considerou-se também importante construir um método que realizasse a operação inversa, isto é, a partir de uma determinada entidade biológica, saber quais são as proteínas ou genes que lhes estão associadas. A Figura 3.22 apresenta o resultado dessa pesquisa.



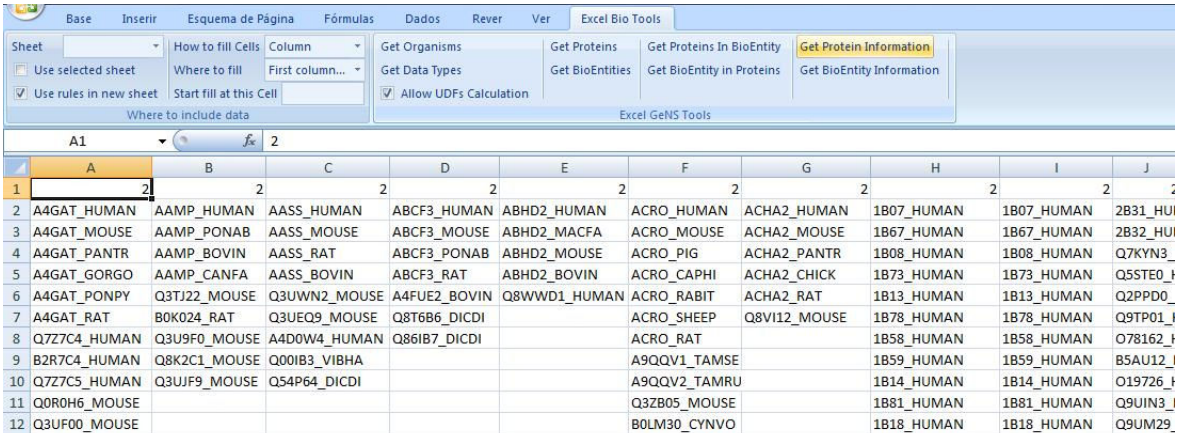
	A	B	C	D	E	F	G	H	I	J	K
1	**Protein Alias**	**Data Type ID**		**Protein Alias**	**Data Type ID**		**Protein Alias**	**Data Type ID**			
2	BC001091	4		AF069067	4		BT006840	4			
3	BC009235	4		AL135903	4		ENSG0000013	27			
4	BC009617	4		AL135903	4		NP_066977.1	6			
5	EHHADH	1		AL135903	4		NP_478059.1	6			
6	Enoyl-CoA hy	132		AL135903	4		PSA	129			

Figura 3.22 - Proteínas que estão envolvidas em determinado processo biológico

Apesar de não ser aqui apresentado, é possível limitar a pesquisa por tipo de dados. Assim, poder-se-á, por exemplo, saber apenas o nome dos genes que estão relacionados com a entidade biológica que está a ser estudada.

No entanto havia ainda uma lacuna a preencher. Os investigadores têm também necessidade de saber quais são as proteínas que determinado gene produz ou realizar a operação inversa.

Os dados apresentados na Figura 3.23 correspondem à execução do método implementado para obter informação entre genes e proteínas. Para este caso em específico, foi realizada uma pesquisa para saber quais são os nomes das proteínas que estão relacionadas com os genes apresentados na coluna E da Figura 3.19 presente na página 67.



	A	B	C	D	E	F	G	H	I	J
1	2	2	2	2	2	2	2	2	2	2
2	A4GAT_HUMAN	AAMP_HUMAN	AASS_HUMAN	ABCF3_HUMAN	ABHD2_HUMAN	ACRO_HUMAN	ACHA2_HUMAN	1B07_HUMAN	1B07_HUMAN	2B31_HU
3	A4GAT_MOUSE	AAMP_PONAB	AASS_MOUSE	ABCF3_MOUSE	ABHD2_MACFA	ACRO_MOUSE	ACHA2_MOUSE	1B67_HUMAN	1B67_HUMAN	2B32_HU
4	A4GAT_PANTR	AAMP_BOVIN	AASS_RAT	ABCF3_PONAB	ABHD2_MOUSE	ACRO_PIG	ACHA2_PANTR	1B08_HUMAN	1B08_HUMAN	Q7KYN3_
5	A4GAT_GORGO	AAMP_CANFA	AASS_BOVIN	ABCF3_RAT	ABHD2_BOVIN	ACRO_CAPHI	ACHA2_CHICK	1B73_HUMAN	1B73_HUMAN	Q5STE0_
6	A4GAT_PONPY	Q3TJ22_MOUSE	Q3UWN2_MOUSE	A4FUE2_BOVIN	Q8WWD1_HUMAN	ACRO_RABIT	ACHA2_RAT	1B13_HUMAN	1B13_HUMAN	Q2PPD0_
7	A4GAT_RAT	B0K024_RAT	Q3UEQ9_MOUSE	Q8T6B6_DICDI		ACRO_SHEEP	Q8VI12_MOUSE	1B78_HUMAN	1B78_HUMAN	Q9TP01_
8	Q7Z7C4_HUMAN	Q3U9F0_MOUSE	A4D0W4_HUMAN	Q86IB7_DICDI		ACRO_RAT		1B58_HUMAN	1B58_HUMAN	O78162_
9	B2R7C4_HUMAN	Q8K2C1_MOUSE	Q00IB3_VIBHA			A9QQV1_TAMSE		1B59_HUMAN	1B59_HUMAN	B5AU12_
10	Q7Z7C5_HUMAN	Q3UJF9_MOUSE	Q54P64_DICDI			A9QQV2_TAMRU		1B14_HUMAN	1B14_HUMAN	O19726_
11	Q0R0H6_MOUSE					Q3ZB05_MOUSE		1B81_HUMAN	1B81_HUMAN	Q9UIN3_
12	Q3UF00_MOUSE					B0LM30_CYNVO		1B18_HUMAN	1B18_HUMAN	Q9UM29_

Figura 3.23 - Integração de dados sobre a própria proteína

Para cada um dos genes apresentados, foi então obtido o nome das proteínas que estão com ele relacionadas. Mais uma vez, a primeira linha funciona como um cabeçalho que indica que toda a coluna contém dados do tipo dois.

Desempenho da ferramenta

Após a construção da ferramenta, foram também realizados alguns testes de desempenho da aplicação que tentam simular casos reais de utilização. Foram incluídos temporariamente "*timers*" para registar os tempos de resposta da aplicação em várias situações.

Na utilização da ferramenta, os utilizadores terão de obter informação sobre o organismo que será alvo da sua análise e sobre todos os tipos de dados existentes na base de dados uma vez que a maioria dos métodos desenvolvidos necessita da identificação taxonómica do organismo e do tipo de dados que o utilizador pretende que seja devolvido para a folha de cálculo. Desta forma, é absolutamente necessário introduzir na folha de cálculo

informação sobre estes dois grupos. Na Tabela 5 está registado o tempo necessário para obter informação sobre os organismos e na Tabela 6 está registado o tempo de acesso a informação sobre os tipos de dados. No consumo do método "*SearchOrganism*" sem parâmetros de entrada são devolvidos dados relativos aos primeiros mil organismos presentes na base de dados. Desta forma, caso se use um parâmetro de entrada, é retornada uma menor quantidade de dados e consequentemente o tempo de acesso e integração de dados diminui. Esta situação também se verifica para o método "*ListDataTypes*".

Tabela 5 – Registo dos tempos de integração a dados sobre organismos

	Tempo em segundos
Sem parâmetros	5,560
Com o parâmetro "h1n1"	0,659
Com o parâmetro "Hsa"	0,411

Tabela 6 – Registo dos tempos de integração a dados sobre tipos de dados

	Tempo em segundos
Sem parâmetros	1,086
Com o parâmetro genename	0,659

Após a introdução destes dados na folha de cálculo, os utilizadores necessitam muitas vezes de obter informação sobre proteínas ou entidades biológicas. Foi registado o tempo de resposta da aplicação para vários casos de utilização. Para as proteínas, foram registados os tempos de resposta para obter uma, dez, cem e mil proteínas de um determinado organismo. Por outro lado, caso sejam utilizadas as UDF, é natural que existam vários pedidos sucessivos de consumo dos seus métodos. Por este motivo, na Tabela 7 estão registados os tempos de consumo dos WebServices.

Tabela 7 - Tempo, em segundos, necessário à integração de dados resultantes de um determinado número de consumos de WebServices (proteínas)

Dados devolvidos	Número de consumos do WebService				
	1	5	10	15	20
1	0,476	1,532	2,765	4,520	5,536
10	0,537	1,616	3,078	3,221	3,427
100	0,569	1,849	1,607	4,280	3,310
1000	0,858	2,276	3,592	4,717	5,791

Para as entidades biológicas os utilizadores não podem definir a quantidade de dados que são devolvidos. Desta forma foram realizados vários pedidos que retornam diferentes quantidades de dados e foram registados os tempos de resposta da aplicação (Tabela 8). Foram também realizados vários pedidos consecutivos uma vez que caso se usem UDF este será o caso típico de utilização.

Tabela 8 - Tempo, em segundos, necessário à execução integração de dados resultantes de um determinado número de consumos de WebServices (entidades biológicas)

Quantidade de dados devolvidos	1 WS (s)	5 WS (s)
1	2,992	12,401
200	3,617	13,448
610	3,387	13,917
16023	5,888	27,388
232118	33,392	2m46s

Como se pode verificar a partir das tabelas, os tempos de resposta da aplicação são aceitáveis uma vez que na maioria dos métodos o tempo de execução é inferior a cinco segundos. No entanto, existem alguns métodos, que por retornarem uma grande quantidade de dados, aumentam o tempo de execução.

Quando é efectuado mais do que um consumo dos métodos dos WebServices, a aplicação apresenta informação de que os pedidos estão a ser efectuados e mostra o progresso existente no menu de estados do Excel. Este menu encontra-se no canto inferior esquerdo

da aplicação. Deste modo, os utilizadores saberão que apesar da aplicação estar a demorar algum tempo a responder, está a apresentar progresso nos cálculos (Figura 3.24).

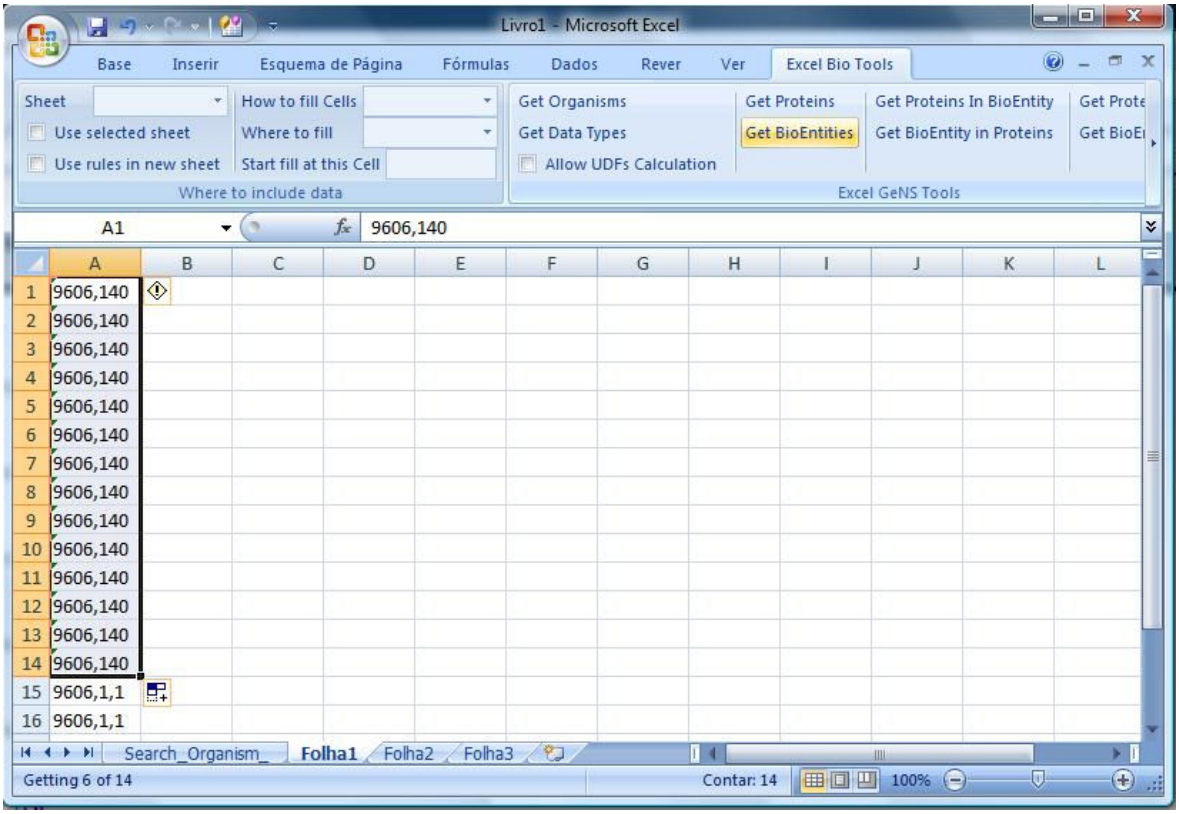


Figura 3.24 - Progresso a efectuar cálculos

A principal razão que levou à apresentação dos tempos de resposta em possíveis cenários de utilização real é comprovar que a aplicação não será demasiado lenta. Caso fosse, os utilizadores poderiam optar por continuar a utilizar as antigas ferramentas. Não é possível determinar todos os casos reais de utilização da ferramenta e desta forma poderão existir pesquisas mais lentas. Por vezes o tempo de execução não apresenta relação com a quantidade de dados que é devolvida pelos WebServices. Esta situação é perfeitamente justificável uma vez que diferentes métodos obtêm dados de diferentes tabelas e realizam diferentes tipos de pesquisas na base de dados. Desta forma, os próprios WebServices poderão ser mais lentos uma vez que a pesquisa realizada na base de dados também o é.

3.4 Sumário

Neste capítulo foi feita uma descrição da abordagem que foi utilizada para cumprir os objectivos predefinidos.

VSTO é uma tecnologia pertencente à plataforma dotNET e é a mais recente tecnologia da Microsoft que permite adaptar o Office às necessidades de cada utilizador. É possível através desta tecnologia construir aplicações fiáveis que permitam manipular documentos do Excel, Word ou outros programas Office. A partir desta tecnologia é possível construir novos menus *"Ribbon",* novas *"User Defined Functions"* para ajudar os utilizadores a realizar as suas tarefas ou documentos com formulários predefinidos.

Utilizaram-se WebServices para mascarar o acesso à base de dados e permitir que de uma forma simples os dados pudessem integrados para o Excel. Para além disso através do uso de WebServices é possível garantir que serão mínimos os problemas de acesso aos dados. Tipicamente, os dados serão integrados directamente para o Excel uma vez que o último método de integração de dados desenvolvido é rápido, fiável e de fácil utilização.

Foram construídas duas interfaces:

- Botões num novo menu "*Ribbon*"
- "*User Defined Functions*" (UDF).

Na verdade as UDF não realizam quase nenhuma operação e foram manipulados os eventos de cálculo do Excel para detectar quando o utilizador introduz uma fórmula nas folhas de cálculo. Na manipulação destes eventos é realizado o processamento que permite utilizar os WebServices e introduzir os dados no Excel.

No grupo "*Where to include data*" os utilizadores poderão definir o local onde desejam incluir os dados que resultam do uso de ambas as interfaces.

Foram implementados métodos que permitem obter informação sobre organismos, genes, proteínas e entidades biológicas. É ainda possível estabelecer relações entre genes e proteínas ou proteínas e entidades biológicas.

4 Conclusões

Neste trabalho foi desenvolvida uma ferramenta que adiciona novas funcionalidades de integração de dados em folhas de cálculo. As novas funcionalidades permitirão que os investigadores utilizem o Excel como uma das suas principais ferramentas para obter dados provenientes de WebServices. Assim, os utilizadores poderão utilizar as folhas de cálculo para organizar, manipular e analisar os dados provenientes de WebServices. Após os dados estarem incluídos no Excel, será possível tirar partido das suas funcionalidades de cálculo matemático ou análise estatística e poder-se-ão criar gráficos.

Tal como estava definido nos objectivos, a integração de dados para o Excel é transparente aos utilizadores. Desta forma, os investigadores podem facilmente incorporar no Excel dados relativos a organismos, proteínas ou processos biológicos. Para além disso, será possível correlacionar dados e determinar, por exemplo, quais os organismos que produzem determinada proteína, qual o gene que é responsável por produzir essa mesma proteína e quais os processos biológicos em que essa proteína está inserida.

Infelizmente, os modelos de programação do Office 2003 e do Office 2007 são significativamente diferentes. A ferramenta desenvolvida apenas adicionará novas funcionalidades ao Office 2007. Esta é uma grande limitação uma vez que ainda existem muitas instituições que apenas possuem a licença do Office 2003.

Assim, pode concluir-se que o principal objectivo deste trabalho foi cumprido. No entanto, no seguimento deste trabalho podem ser desenvolvidas novas funcionalidades para o Excel. Neste momento, a aplicação desenvolvida usa o GeNS como principal fonte de dados. Existe na ferramenta a possibilidade de consumir outros WebServices. No entanto, este consumo não é trivial e é necessário conhecer-se o URL completo que já inclua os parâmetros de consumo.

No futuro, poderão ser adicionados métodos que permitam consumir WebServices ou aceder a bases de dados de outros sistemas. Assim, o Excel poderá vir a ser utilizado como integrador de diferentes fontes de dados e tal como no trabalho desenvolvido aproveitar-se-ia uma interface já existente para apresentar dados que devido à sua natureza necessitam de ser organizados e analisados.

As interfaces construídas são muito pouco flexíveis. Caso se desejem integrar novos métodos aos WebServices ou mesmo alterar os métodos existentes, será necessário proceder a algumas alterações na ferramenta construída. Na opinião do autor, esta é a maior lacuna da aplicação implementada. Desta forma, devem ser estudadas formas que permitam que a interface se adapte às condições fornecidas pelos WebServices.

5 Referências

[1] Butler K. clickstreamtech.com. [Online].; 2008 [cited 2009 6 20. Available from: http://www.clickstreamtech.com/11.14.08.html.

[2] Shepherd G. Microsoft ASP.NET 2.0 Step by Step Washington: Microsoft Press; 2005.

[3] Bruney A. Professional VSTO 2005 Visual Studio 2005 Tools for Office Indiana: Wrox; 2006.

[4] Nagel C, Evjen B, Glynn J, Watson K, Skinner M. Professional C# 2008 Indiana: Wrox; 2008.

[5] msdn.microsoft.com. msdn.microsoft.com. [Online]. [cited 2009 6 20. Available from: http://msdn.microsoft.com/en-us/library/8553caee.aspx.

[6] msdn.microsoft.com. msdn.microsoft.com. [Online]. [cited 2009 6 20. Available from: http://msdn.microsoft.com/en-us/library/15s06t57.aspx.

[7] Watson K, Nagel C, Pedersen JH, Reid JD, Skinner M, White E. Beggining Microsoft Visual C# 2008 Indiana: Wrox; 2008.

[8] msdn.microsoft.com. msdn.microsoft.com. [Online]. [cited 2009 6 20. Available from: http://msdn.microsoft.com/en-us/library/a2c7tshk.aspx.

[9] msdn.microsoft.com. msdn.microsoft.com. [Online]. [cited 2009 6 20. Available from: http://msdn.microsoft.com/en-us/library/12a7a7h3.aspx.

[10] msdn.microsoft.com. msdn.microsoft.com. [Online]. [cited 2009 6 20. Available from: http://msdn.microsoft.com/en-us/library/0xy59wtx.aspx.

[11] Richter J. msdn.microsoft.com. [Online]. [cited 2009 6 20. Available from: http://msdn.microsoft.com/en-us/magazine/bb985010.aspx.

[12] msdn.microsoft.com. msdn.microsoft.com. [Online]. [cited 2009 6 20. Available from: http://msdn.microsoft.com/en-us/library/bdts8hk0(VS.71).aspx.

[13] Northup T, Wildermuth S, Ryan B. Microsoft.NET Framework 2.0 Application Development Foundation Redmond, Washington: Microsoft Press; 2006.

[14] msdn.microsoft.com. msdn.microsoft.com. [Online]. [cited 2009 6 20. Available from: http://msdn.microsoft.com/en-us/library/kbcw921f(VS.80).aspx.

[15] Pereira J. Bioinformatics.ua.pt. [Online].; 2009 [cited 2009 6 20. Available from: http://bioinformatics.ua.pt/applications/gens.

[16] Cornell P. msdn.microsoft.com. [Online].; 2002 [cited 2009 06 20. Available from: http://msdn.microsoft.com/en-us/library/aa140260(office.10).aspx.

[17] Brown M. blogs.msdn.com. [Online].; 2008 [cited 2009 6 20. Available from: http://blogs.msdn.com/mab/archive/2008/01/02/soap-toolkit-vista-excel-2007-call-web-services-impossible.aspx.

[18] Haddad R. www.microsoft.com. [Online]. [cited 2009 6 20. Available from: http://www.microsoft.com/brasil/msdn/Tecnologias/vsnet/OfficeVSTO2Excel.mspx.

[19] Khen D, Prish S. msdn.microsoft.com. [Online].; 2007 [cited 2009 6 20. Available from: http://msdn.microsoft.com/en-us/library/bb226683.aspx.

[20] Carter E. blogs.msdn.com. [Online].; 2004 [cited 2009 6 20. Available from: http://blogs.msdn.com/eric_carter/archive/2004/12/01/273127.aspx.

[21] Carte E, Lippert E. Visual Studio Tools for Office: Using C# with Excel, Word,

Outlook, and InfoPath: Addison Wesley Professional; 2005.

[22] Green J, Bullen S, Bovey R, Alexander M. Excel 2007 VBA Programmers Reference Indianapolis, Indiana: Wrox; 2007.

[23] Gunderloy M. msdn.microsoft.com. [Online].; 2002 [cited 2009 6 20. Available from: http://msdn.microsoft.com/en-us/library/ms973802.aspx.

[24] Gunderloy M. msdn.microsoft.com. [Online].; 2001 [cited 2009 6 20. Available from: http://msdn.microsoft.com/en-us/library/ms973800.aspx.

[25] Khen D, Prish S. msdn.microsoft.com. [Online].; 2007 [cited 2009 6 20. Available from: http://msdn.microsoft.com/en-us/library/bb226682.aspx.

[26] Thomas C. wrox.com. [Online]. [cited 2009 6 20. Available from: http://www.wrox.com/WileyCDA/Section/Excel-Services-User-Defined-Functions-UDFs-.id-305100.html.